

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Infographie : Le réalisme des images de synthèse

Arcuri, Antonio

Award date:
1991

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix,
Institut d'Informatique,
21 rue Grandgagnage,
5000 NAMUR

INFOGRAPHIE :

Le réalisme des images de synthèse.

Antonio ARCURI

Promoteur : Baudouin LE CHARLIER.

Mémoire présenté en vue
de l'obtention du titre de
Licencié et Maître en
Informatique.

Année Académique 1990-1991

Résumé

Auparavant, le terme *infographie* évoquait souvent les simples tracés de graphes ou de schémas en deux dimensions. Aujourd'hui, l'infographie et la photographie semblent coïncider... L'infographie permet à présent de créer et de manipuler des images couleurs en trois dimensions avec un réalisme photographique. Que ce soit dans les domaines de la création, de la conception, ou de l'information, le réalisme des images de synthèse impressionne. Pour créer des images de synthèse réalistes, deux phases sont nécessaires : la modélisation et le rendu réaliste. La première consiste à modéliser des objets tridimensionnels, les disposer dans l'espace de manière à former une scène 3D, et définir enfin la position de l'observateur de cette scène. La seconde phase consiste à mettre en oeuvre des techniques qui permettent d'*habiller* cette scène, d'y ajouter du réalisme, afin de produire des images non discernables de photographies d'objets réels. Une première étape essentielle consiste à éliminer les parties cachées de la scène perçue par l'oeil d'un observateur. La seconde étape tente d'apporter des effets d'illumination sur les objets de la scène, de simuler des ombres, en tenant compte des transparences éventuelles. La dernière étape du rendu réaliste ajoute éventuellement des textures pour simuler des surfaces irrégulières. L'algorithme du lancer de rayons permet de tenir compte de toutes ces étapes, mais au prix d'un temps calcul considérable.

Abstract

In the past, the term *computer graphics* implied line graphs and bar charts in two dimensions. Now, graphics and photography seem to have merged. Today, computer graphics can create and manipulate color images in three dimensions with photographic precision and photorealism. Either in the fields of creation, conception, or information, the realistic image synthesis forces our admiration. Two stages are necessary : modeling and rendering. In the first stage, tridimensionnal objects are modeled, then they are placed so that they can form a 3D scene, and an observer's position finally can be defined. In the second stage, the techniques which allow to fill the scene and give realism are carried out. The aim of the rendering techniques is to produce images which can not be distinguished from photographs of real objects. A first essential step determines the visible surfaces of the scene's objects viewed by the observer. The second step tries to bring lighting effects on the objects of the scene, and simulate shadows, while taking possible transparencies into account. The last step of rendering may add textures in order to simulate irregular surfaces. Ray-tracing algorithm can render all this effects, but with a notable processing time.

Ma reconnaissance toute particulière s'adresse à Monsieur le Professeur B. Le Charlier pour avoir accepté la direction de ce mémoire et pour la confiance qu'il a placée en moi dans le choix et la réalisation de ce travail.

Je tiens ensuite à remercier Monsieur le Professeur D. Thalmann, pour son accueil au sein de son équipe de recherche du laboratoire d'infographie de l'Ecole Polytechnique Fédérale de Lausanne. J'aimerais remercier particulièrement tous les membres du LIG qui m'ont aidé, grâce à leurs connaissances étendues dans ce domaine, à aborder les techniques infographiques ; Francis Balaguer, Ronan Boulic, Enrico Gobbetti, André LeBlanc, Fernando Matomira, Angelo Mangili, Olivier Renault, et Russell Turner.

Mes remerciements s'adressent également à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

Ce mémoire marque la fin d'un long cycle d'études. Je remercie mes parents, sans lesquels je n'aurais pu l'entreprendre.

Namur, le 30 Août 1990
Antonio ARCURI

Table des matières

<u>CHAPITRE I. INTRODUCTION</u>	1
<u>CHAPITRE II. EVOLUTION DE L'INFOGRAPHIE</u>	3
II.1 Définition	4
II.2 Historique.....	5
II.3 Intérêt et domaines d'application	8
II.4 Conclusion.....	12
<u>CHAPITRE III. LA SYNTHÈSE DE SCÈNES 3D</u>	13
III.1. Introduction	14
III.2 La modélisation 3D	15
III.2.1 Introduction	15
III.2.2 La géométrie constructive.....	16
III.2.3 Le modèle polyédrique	18
III.2.4 Les courbes et surfaces courbes	22
III.2.5 Les modèles "évolués"	23
III.3 Les transformations et la visualisation 3D	24
III.3.1 Introduction	24
III.3.2 L'espace objet et l'espace image.....	25
III.3.3 Les transformations géométriques	26
III.3.4 Les projections géométriques.....	29
III.4 Conclusion.....	30
<u>CHAPITRE IV. LE RENDU RÉALISTE DE SCÈNES 3D</u>	31
IV.1 Introduction	32
IV.2 Les étapes du rendu réaliste.....	32
IV.3 L'élimination des parties cachées	33
IV.3.1 Introduction	33
IV.3.2 Classification des algorithmes.....	34
IV.3.3 L'algorithme de Z-Buffer	34
IV.3.4 L'algorithme de lancer de rayons.....	35

IV.3.5 Conclusion	36
IV.4 Les modèles d'illumination	37
IV.4.1 Introduction	37
IV.4.2 La lumière synthétique	37
IV.4.3 Le modèle de Phong et Blinn.....	39
IV.4.4 Le modèle d'illumination simplifiée.....	47
IV.4.5 Le modèle d'illumination totale.....	49
IV.4.6 Le modèle de Cook	50
IV.4.7 Le modèle théorique de Torrance et Sparrow.....	51
IV.4.8 La répartition de la lumière sur les surfaces	51
IV.4.9 Conclusion	58
IV.5 La transparence	58
IV.5.1 La méthode du peintre.....	59
IV.5.2 La loi de Snell-Descartes	59
IV.6 La détermination des ombres	60
IV.6.1 Les types d'ombres	60
IV.6.2 Les techniques de production d'ombres portées	61
IV.7 Les textures.....	64
IV.7.1 Les techniques de textures	64
IV.7.2 L'application d'une image 2D.....	65
IV.7.3 La perturbation de la lumière	66
IV.8 Le lancer de rayons	67
IV.8.1 Introduction	67
IV.8.2 Le parcours des rayons lumineux.....	67
IV.8.3 Le principe.....	68
IV.8.4 L'algorithme du lancer de rayons.....	69
IV.8.5 Les tests d'intersection	71
IV.8.6 La direction des rayons réfléchis et incidents	72
IV.8.7 Le calcul de l'intensité lumineuse	72
IV.8.8 Conclusion	72
IV.9 La radiosité.....	73
IV.10 Conclusion	73

CHAPITRE V. ED3D : UN EDITEUR HIERARCHIQUE DE SCENES 3D REALISTES74

V.1 Introduction	75
V.2 Le lieu de stage	75
V.3 L'environnement de travail	76

V.3.1 La "POWER Iris"	77
V.3.2 La "GL" de l'Iris 4D-GT	81
V.3.3 Conclusion	94
V.4 Présentation du logiciel ED3D	94
V.4.1 Introduction	94
V.4.2 Objectifs du logiciel ED3D	95
V.4.3 Les concepts d'ED3D	95
V.4.4 La structure hiérarchique	96
V.4.5 Le modèle conceptuel des données	100
V.4.6 Spécifications fonctionnelles	102
V.4.7 Evolution du logiciel	107
V.4.8 Commentaires sur l'implémentation.....	109
V.5 Conclusion.....	109
<u>CHAPITRE VI. CONCLUSION</u>	110
<u>CHAPITRE VII. BIBLIOGRAPHIE</u>	112

I. INTRODUCTION

Les images produites par ordinateur ont envahi notre vie quotidienne. Il suffit de regarder la plupart des génériques de télévision ou quelques séquences de publicité pour s'en convaincre...

Les progrès de ces dernières années ont été surtout réalisés dans les *techniques de rendu réaliste* des images de synthèse. Ces nouvelles techniques permettent d'envisager à présent l'usage d'illustrations aussi réalistes que des photographies. Il est donc naturel que de nombreuses personnes souhaitent comprendre comment ces images sont produites, qu'il s'agisse de l'amateur désireux de connaître les grandes lignes des traitements à réaliser, ou de l'informaticien voulant développer une nouvelle application utilisant des techniques infographiques.

Ce mémoire tente de répondre à cette demande en présentant les méthodes qui permettent de créer des images dont le réalisme approche celui d'une photographie. Il est structuré de la manière suivante :

Le chapitre II présente tout d'abord un bref historique de l'infographie et son intérêt dans les nombreux domaines où elle peut s'appliquer.

Deux phases sont nécessaires à la création d'images de synthèse réalistes : le *modeling* et le *rendering*.

La première phase sera traitée dans le chapitre III. Il est en effet indispensable, dans cette phase de *modeling*, d'élaborer le modèle mathématique des objets à visualiser. Une fois les objets modélisés, ils sont disposés dans l'espace de manière à composer une *scène 3D*. On spécifie finalement comment cette scène est vue par un observateur particulier, afin de déterminer la projection géométrique à opérer sur le modèle 3D de la scène et d'obtenir ainsi une image en deux dimensions.

Nous ne nous attarderons pas sur la phase de modélisation - qui mériterait un dossier à part - mais plutôt sur les techniques de rendu réaliste, qui feront l'objet du chapitre IV.

Dans ce chapitre IV, nous nous poserons la question suivante : Comment rendre une scène 3D *réaliste*? Nous examinerons toute une série d'étapes qui peuvent être mises en oeuvre pour y arriver ; l'élimination des parties cachées, les modèles d'illumination, les effets spéciaux (ombres, transparences, textures des surfaces,...), ou le célèbre algorithme de *lancer de rayons*.

Pour maîtriser les techniques infographies présentées dans les chapitres III et IV, il est certain qu'une réalisation pratique était de mise.

Une scène 3D est fréquemment composée de divers objets, parfois complexes. Ces objets complexes sont alors formés de sous-objets plus simples, qui sont eux-mêmes décomposés jusqu'à atteindre les primitives graphiques de base. On façonne de la sorte une structure hiérarchique, qui peut servir de base à un système d'animation de la scène. On voit donc l'intérêt d'un outil permettant d'éditer de manière interactive des scènes 3D réalistes, et qui tient compte de ce concept de hiérarchie. C'est ainsi que je présenterai, dans le chapitre V, une implémentation personnelle réalisée pendant la période de stage au laboratoire d'infographie de l'Ecole Polytechnique Fédérale de Lausanne : un éditeur hiérarchique de scènes 3D réalistes.

Ce mémoire n'a pas la prétention de présenter l'état de l'art dans ce vaste domaine, mais de susciter, je l'espère, l'intérêt du lecteur pour l'imagerie de synthèse réaliste. Le spécialiste désireux d'en savoir plus, pourra prolonger cette entrée en matière grâce aux nombreuses références bibliographiques.

II. EVOLUTION DE L'INFOGRAPHIE

II.1 Définition

Le terme *infographie* a été formé en 1974 par le rapprochement des mots *informatique* et *graphique* pour servir de traduction à l'expression anglaise "Computer Graphics". Cette branche de l'informatique englobe les domaines et les applications utilisant l'ordinateur pour créer, mémoriser et manipuler des graphiques, dessins, schémas ou images. L'infographie est souvent qualifiée d'*interactive*. En effet, ses applications sont principalement interactives, c'est-à-dire supportant un dialogue avec l'utilisateur et permettant, par une interaction continue, d'affiner progressivement les images affichées par le biais de périphériques graphiques.

A priori, la **synthèse d'images** rejoint le **traitement d'images** ("image processing"), car dans les deux cas on travaille principalement sur des images vidéo. La synthèse d'images repose sur des processus de génération de structures de données représentant un ou plusieurs objets formant une *scène*. Des manipulations (telles des transformations géométriques, ...) sur ces structures de données se traduisent par des modifications de l'aspect physique de l'image vidéo résultante. Le traitement d'images traite le processus inverse : l'analyse de scènes réelles ou la reconstruction du modèle bidimensionnel ou tridimensionnel d'objets d'une scène à partir d'une image vidéo de celle-ci. Le traitement d'images est particulièrement important dans les domaines de la photogrammétrie et la télédétection, la surveillance aérienne, la robotique, l'imagerie médicale, les analyses d'empreintes, la reconnaissance de caractères.

L'infographie concerne la synthèse picturale d'objets réels ou imaginaires à partir de leurs modèles informatiques. Ainsi, le traitement d'images doit être considéré comme une discipline distincte de l'infographie, dérivée du traitement numérique des signaux, et regroupant les techniques de reconnaissance de formes, d'analyse, de filtrage, de rehaussement, de codage et de compression d'images. La figure II.1 [LIEB 88] schématise les étapes de la synthèse et du traitement d'images.

II. Evolution de l'infographie

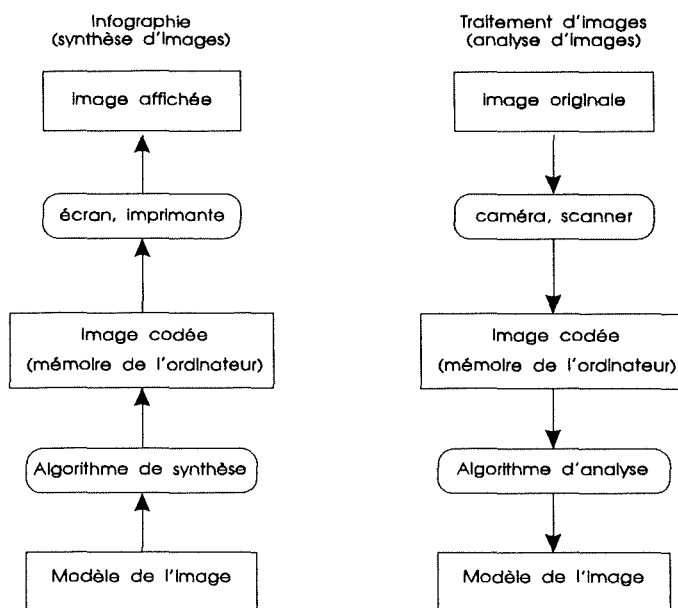


Figure II.1 : Infographie vs traitement d'images.

L'infographie se définit donc comme l'utilisation de l'ordinateur pour manipuler des dessins, images, et autres objets graphiques, à la restriction du traitement d'images. Elle peut aussi être réciproquement vue comme l'utilisation de dessins et d'images pour représenter des données ou des résultats provenant de traitements informatiques. Tout simplement, on peut reprendre la définition de l'informatique en y ajoutant l'adjectif graphique [SCHW 87]:

- traitement graphique de l'information, et
- traitement de l'information graphique.

II.2 Historique¹

Avant d'entamer cet historique, il faut tout d'abord souligner le fait que l'évolution du matériel dédié au graphisme a eu une influence primordiale sur l'évolution du logiciel graphique. En effet, la faisabilité de certaines applications infographiques dépend directement des performances du matériel disponible.

C'est en 1950 que le M.I.T. ("Massachusetts Institute of Technology") développe un **écran graphique vectoriel** composé d'un tube cathodique contrôlé directement par ordinateur (le "Whirlwind Computer"). Ce type d'écran permet d'afficher directement des

¹Références bibliographiques principales : [LIEB 88], [SCHW 87] et [FOLE 90].

II. Evolution de l'infographie

vecteurs. La *mémoire de rafraîchissement de l'écran* contient alors une liste des vecteurs à afficher périodiquement, avec les coordonnées des points extrêmes et leur intensité Noir/Blanc (voir figure II.2 extraite de [FOLE 90]). Bien que ne permettant que des dessins simples, ce dispositif servit de modèle aux systèmes développés durant les quinze années suivantes.

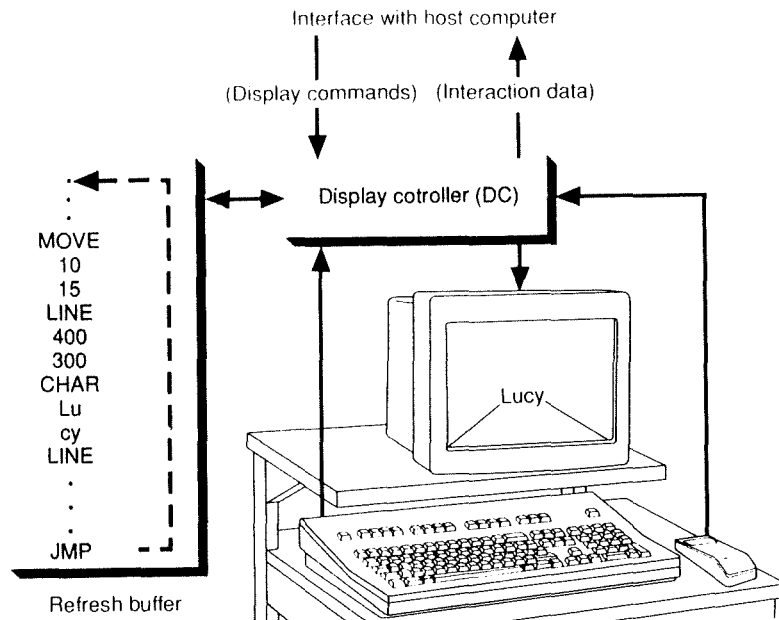


Figure II.2 : Le principe d'un écran graphique vectoriel.

Un autre point clé des débuts de l'infographie interactive fut le travail d'Ivan Sutherland (dans le cadre de sa thèse de doctorat au M.I.T.) sur son système de dessin **Sketchpad** en 1963. Cette époque connut le développement des systèmes de dessin assisté par ordinateur chez General Motor™, Lockheed Aircrafts™ et Bell Telephone™ principalement. Apparurent également les premiers systèmes de conception assistée par ordinateur dans le domaine du dessin technique. Divers projets de recherche ont également vu le jour.

La commercialisation d'un terminal graphique de haute qualité à un prix attractif par Tektronix™ (modèle 4012-4014) vers 1970 permit le véritable départ de l'infographie dans l'industrie. Ce terminal, doté d'un **écran vectoriel à mémoire** (écran vectoriel mémorisant chaque vecteur jusqu'à l'effacement global) devint le premier poste de travail graphique.

L'abaissement du prix des mémoires permit ensuite de réaliser des **écrans matriciels** (appelés aussi "ponctuels") utilisant une technologie dérivée des terminaux alphanumériques. L'image est alors créée sur écran point par point à partir d'une mémoire de rafraîchissement où chaque bit correspond à un point de l'écran (voir figure II.3 extraite de [FOLE 90]).

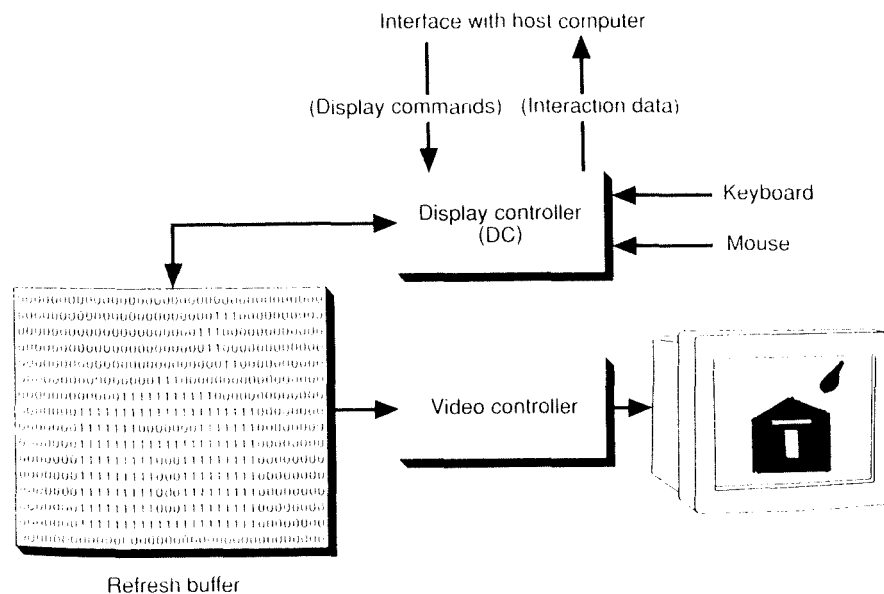


Figure II.3 : Le principe d'un écran graphique matriciel.

La couleur, disponible depuis longtemps sur les traceurs, est ensuite apparue sur les écrans, grâce à la commercialisation d'un **tube cathodique couleur à masque** (utilisé dans les récepteurs de télévision). La plupart des écrans actuels sont toujours basés sur cette technologie. De nombreux périphériques d'entrée furent ensuite développés, dont la souris qui est remarquable par sa simplicité et son ergonomie. Dès l'apparition des micro-ordinateurs en 1977, des cartes d'extension graphique furent proposées. Les générations suivantes ont enfin incorporé une interface graphique de manière standard.

Face à ce développement des logiciels utilisant du graphisme, naquit le désir de standardiser les bibliothèques graphiques pour permettre une portabilité des applications. Plusieurs propositions sont apparues, dont émergent CORE [GSPC 79] et, plus tard, GKS ("Graphical Kernel System") [ANSI 85b] et son extension GKS-3D [INTE 88], mais surtout PHIGS ("Programmer's Hierarchical Interactive Graphics System") [ANSI 88].

Vers 1982, se développent les stations de travail graphiques, qui grâce aux réseaux locaux, constituent un véritable environnement de travail intégré. On assiste aussi à l'épanouissement de la Conception Assistée par Ordinateur (la CAO). Ces systèmes deviennent une alternative économique à l'informatique traditionnelle basée sur des ordinateurs centraux. En 1985, la démocratisation et la décentralisation de l'informatique et du graphisme se poursuit, précédée par la diminution du prix du matériel. Les logiciels de CAO dépassent alors largement, en terme de prix et de complexité, les configurations matérielles qu'ils nécessitent et ne peuvent être rentabilisés que par une large diffusion et

conçus pour l'utilisateur et non plus pour le spécialiste.

La génération actuelle des machines haut de gamme se compose de stations graphiques professionnelles munies d'un écran couleur haute définition, d'importantes ressources de calcul et de mémoire, comparables à celles d'un ordinateur central de la génération antérieure. Elles disposent généralement d'une importante bibliothèque graphique et permettent de réaliser des images aussi réalistes que des photographies. Des processeurs ou coprocesseurs graphiques spécialisés sont souvent incorporés à ces machines sous la forme de circuits intégrés à large échelle (les circuits VLSI - "Very Large Scale Integration").

Les catégories traditionnelles perdent à présent leur signification ; on en arrive à classer les systèmes en fonction de leur application et non plus de leur matériel.

II.3 Intérêt et domaines d'application

L'intérêt du graphisme a été reconnu très tôt ; cela tient du proverbe chinois : "une image vaut mille mots". L'image permet un plus grand débit d'information que le texte, car elle est comprise globalement et non séquentiellement. La quantité d'information y est très élevée, la *vue éclatée* d'un plan d'une maison en étant un exemple type. De plus, il est possible de jouer sur de nombreux paramètres facilement identifiables : la forme, l'intensité, la couleur, la texture... Ces paramètres peuvent prendre des valeurs différentes en tout point de la surface de l'image et évoluer dans le temps. L'information sous forme graphique est également plus claire ; elle peut être interprétée plus rapidement, car elle nécessite un décodage moins complexe que le texte.

L'infographie peut s'appliquer dans presque tous les domaines scientifiques et techniques concernés par l'informatique, et surtout dans des domaines où l'informatique traditionnelle n'avait pas réussi à s'imposer. Ces champs d'application s'étendent avec l'augmentation des performances et la diminution des prix du matériel.

Je citerai ci-dessous quelques domaines privilégiés de l'infographie. Il s'agit cependant d'une liste composée sans souci d'exhaustivité, l'imagerie de synthèse étant en telle phase d'expansion, que tout essai dans ce sens eut été futile.

II.3.1 Le tracé de graphes

Le tracé de graphes constitue l'utilisation la plus simple de l'infographie. Les graphiques ont toujours été préférés à des listes de valeurs pour observer l'évolution spatiale

ou temporelle d'un phénomène complexe. On peut arriver, grâce aux nombreux types de graphes, à des représentations très soignées.

II.3.2 Le Dessin Assisté par Ordinateur (DAO)

Le dessin est lié intimement aux activités de conception dans tous les domaines. L'édition interactive de dessins ou de schémas permet de concevoir plus rapidement, de corriger facilement, et même de répercuter automatiquement, sur tous les éléments dépendants, l'effet d'une modification d'un paramètre. L'ordinateur offre ici les mêmes avantages que dans le traitement de texte : facilité de conception, précision et qualité de l'impression, conduisant à une augmentation de productivité et de qualité.

Les systèmes de dessins techniques, arrivés à maturité, se retrouvent en mécanique, en architecture, en cartographie, en électronique. Le dessin artistique se développe beaucoup actuellement. La différence entre dessin technique et artistique n'est pas évidente, car l'aspect extérieur d'un produit industriel est non seulement dépendant d'impératifs techniques, mais aussi de critères esthétiques. La publicité, par exemple, fait un grand usage d'images de synthèse, de plus en plus réalistes.

II.3.3 La Conception Assistée par Ordinateur (CAO)

Le raffinement et le développement des techniques de DAO conduisit naturellement à informatiser toutes les phases de la conception d'un produit. La CAO désigne la création et la modification d'une structure de données, représentant le modèle d'un objet, par interaction graphique, et par l'usage d'un ensemble de facilités et d'automatismes. Des systèmes experts, basés sur les techniques d'"intelligence" artificielle, commencent à s'intégrer à ces outils. Par exemple, de tels systèmes sont utilisés en mécanique, pour le calcul et la simulation de résistance des matériaux, de contraintes thermiques, de phénomènes de vibration,...

Le génie civil et l'architecture offrent un champ d'action privilégié à la CAO. Elle est devenue également nécessaire dans les domaines où le niveau de complexité atteint est tel qu'il n'est plus possible d'être géré par l'être humain. C'est le cas notamment dans le travail de conception des circuits intégrés actuels, avec par exemple un million de transistors pour un micro-processeur 32 bits. On le perçoit, des outils de conception et de simulation sont indispensables pour gérer les problèmes de placement, de routage, d'optimisation combinatoire.

II.3.4 La Fabrication Assistée par Ordinateur (FAO)

La FAO est l'aboutissement de la CAO. En effet, les systèmes de CAO actuels vont

jusqu'à incorporer la gestion de projet et aboutissent à la fabrication automatique d'ébauches tridimensionnelles. La puissance de cette assistance réside dans son intégration sur toute la chaîne de production industrielle. L'utilisation de la FAO interactive vise non seulement une meilleure productivité, mais également une plus grande souplesse de l'appareil de production, permettant de satisfaire le désir du client et rompant avec la standardisation des produits.

II.3.5 L'Enseignement Assisté par Ordinateur (EAO)

L'infographie permet, dans cet important domaine de recherche, un dialogue plus aisé. Ainsi, pour l'enseignement de l'algèbre linéaire ou de l'histoire de l'art, il est plus efficace de visualiser, avec divers degrés de réalisme, le concept étudié (une surface quadrique ou le style architectural de la Renaissance). Grâce aux systèmes d'animation d'images et aux techniques d'"intelligence" artificielle, on peut développer de vrais didacticiels et "pédagogiciels" tenant compte des connaissances et des aptitudes des élèves.

II.3.6 La Publication Assistée par Ordinateur (PAO)

Même une activité comme le traitement de texte, qui par sa nature n'implique pas de graphique, peut s'envisager de manière graphique, et conduire à un outil plus souple, plus agréable, tel que l'a démontré Apple™ avec son micro-ordinateur "Macintosh". Les possibilités d'intégrer textes et images ont ouvert de nouvelles perspectives pour les éditeurs, et lancé la Publication Assistée par Ordinateur.

II.3.7 Le contrôle de processus

De nombreux systèmes sont soit naturellement instables, soit dépendants d'autres systèmes. Pour garantir un fonctionnement sûr de ce type de système, en plus de leur automatisation, on doit pouvoir être informé et superviser l'état de celui-ci, comme par exemple lors de la surveillance des éléments critiques d'une centrale nucléaire. A partir d'une représentation graphique de cet état, on peut prendre des décisions rapidement et agir sur les commandes du système.

II.3.8 La simulation

On est très proche du contrôle de processus. Ici, pour montrer l'état d'un système, on utilise de plus en plus souvent des techniques de synthèse d'images. La simulation de phénomènes complexes peut alors être visualisée avec une telle précision qu'il est difficile de la distinguer de la réalité. L'exemple le plus classique, et non le plus simple, est le simulateur de vol. D'autres exemples ne manquent pas : la simulation numérique des écoulements d'une turbine, la croissance et la reconstruction de structures cristallines, la

simulation stochastique de processus industriels, la simulation d'opérations chirurgicales, la simulation de "docking" moléculaire, la déflexion d'une aile d'avion lors d'un vol supersonique, la déformation d'un faciès humain selon l'âge,...

II.3.9 L'interface homme-machine

On a créé grâce à l'usage de l'infographie la métaphore de *bureau*, les notions d'icônes, et les systèmes de multi-fenêtrage. Tous les types d'applications ont à présent repris ces techniques. L'utilisation de moyens d'entrée graphique bidimensionnelle (2D) et tridimensionnelle (3D) est plus récente, elle offre des avantages indéniables pour les opérations courantes de choix dans un menu, de confirmation, de désignation et de manipulation tridimensionnelle d'objets.

II.3.10 Les jeux vidéo

On peut arriver, grâce aux techniques de pointe, à concevoir des jeux vidéo très sophistiqués. Des disques vidéo numériques sont utilisés pour le mixage d'images de synthèse et de séquences réelles filmées.

II.3.11 Les systèmes d'animation

Les producteurs de dessins animés se sont intéressés aux techniques d'animation d'images de synthèse. Les publicistes emploient également de plus en plus ces techniques pour créer et animer leurs logos. De nombreux films d'animation ont été produits et ont impressionné par leur réalisme et/ou leur qualité artistique. De nombreuses recherches sont également effectuées dans le domaine de l'animation d'acteurs synthétiques et la simulation de modèles de marche de l'être humain.

II.3.12 La "réalité virtuelle"

Le réalisme des images de synthèse actuelles et les nouveaux périphériques permettent à présent de représenter des univers imaginaires en leur donnant toutes les apparences du réel, ou de reconstruire des mondes artificiels pour la simulation comme pour les jeux de rôle. Il est par exemple possible de porter des lunettes *digitales* et de se balader dans un univers tout à fait programmé !

II.4 Conclusion

On le perçoit, les domaines où les techniques infographiques peuvent intervenir sont nombreux. Le développement du matériel devrait encore renforcer l'intégration et l'utilisation des images de synthèse. Le nombre d'exemples n'a en fait qu'une seule limite, la créativité des infographistes. Une chose est sûre ; dans tous ces domaines, on tente de plus en plus à produire le maximum de réalisme.

III. LA SYNTHESE DE SCENES 3D

III.1 Introduction

Afin de percevoir notre environnement, nous nous servons de nos cinq sens : la vue, le toucher, l'ouïe, le goût et l'odorat. Ceux-ci nous transmettent des informations physiques et chimiques qui nous renseignent sur la forme, la nature, l'aspect, la distance des objets qui forment la scène que l'on observe. Mais pour nous apercevoir réellement de ce qui nous entoure, il nous est nécessaire de nous déplacer tout autour, ou de déplacer les objets eux-mêmes... Nous tournons ainsi un nouvel objet dans nos mains pour mieux nous rendre compte de ses caractéristiques. Enfin, nous mémorisons cette scène ou nous concrétisons une imagination en réalisant des *images*. Ces images mémorisées nous permettent de *reconnaître* une scène de notre vie quotidienne ou un objet imaginaire.

Il est donc important de pouvoir modifier son point de vue et observer la scène dans différentes positions. C'est par ces mouvements que l'interprétation d'une scène est rendue plus facile. La représentation graphique d'une partie de l'espace (3D) sous la forme d'une image dans le plan (2D) est alors obtenue au moyen d'une projection géométrique. Les mécanismes de perception visuelle de l'être humain, les techniques de peinture et de la photographie reposent également sur ce principe de projection. Toutes ces manipulations (déplacements d'objets, projection,...) peuvent être formulées mathématiquement et simulées sur ordinateur.

Par *scène tridimensionnelle*, il faut entendre un type de description consistant à construire une structure de données associée à une modélisation dans l'espace tridimensionnel [BRET 88]. Un objet de la scène sera décrit dans cet espace en terme de primitives géométriques tels que segments de droite et courbes, et dont la plus simple est le point. Ce dernier sera représenté par un triplet de nombres réels, symbolisant la coordonnée tridimensionnelle (x, y, z) de ce point dans l'espace. Ce modèle géométrique est ensuite "regardé" par l'oeil d'un observateur. En fonction de la position de cet oeil et de certains paramètres de visualisation, une projection est ensuite calculée et l'image plane qui en résulte est finalement affichée. Différents types de projections peuvent être envisagés, allant de la symbolique projection orthogonale aux multiples types de projections perspectives (correspondant à notre vision naturelle). La notion de **transformation géométrique** est donc essentielle pour la modélisation et la visualisation, puisqu'elles permettent non seulement de manipuler les objets de la scène, mais également de visualiser celle-ci en 2D.

Plutôt que de donner une schématisation complète des constructions des différentes transformations géométriques, je présenterai dans ce chapitre l'usage de ces outils tout à fait indispensables en infographie. Mais auparavant, une importante étape est nécessaire ; la modélisation mathématique 3D de la scène.

III.2 La modélisation 3D

III.2.1 Introduction

Un volume (ou une surface) est un ensemble infini de points géométriques ; il est donc impossible de mémoriser l'ensemble de ces points. Un volume ne peut donc être modélisé que moyennant un certain nombre d'hypothèses quant à sa structure. Ces hypothèses permettent d'en construire une représentation à partir d'un nombre fini de paramètres. On approche dès lors le volume par un nouveau volume simplifié, déterminé par un nombre limité de points.

Pour décrire la position de ces points dans l'espace, il est nécessaire de se donner un repère. Ce dernier consiste en trois axes OX , OY , et OZ , perpendiculaires deux à deux et passant par le même *point origine* O . Les plans XY , XZ , et YZ sont définis comme passant par les axes désignés. Soit x la distance d'un point P au plan YZ , y la distance de P au plan XZ et z la distance au plan XY . Les trois nombres x , y , et z sont appelés les *coordonnées* du point P . Ces nombres déterminent la position de P dans l'espace tridimensionnel. Le repère $OXYZ$ est supposé fixé dans cet espace². Pour exprimer que x, y et z sont les coordonnées de P , on les écrit couramment sous la forme du triplet de nombres (x, y, z) , comme représenté à la figure III.1.

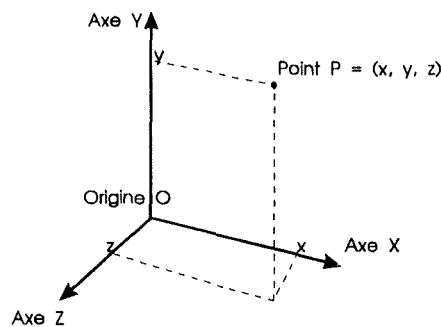


Figure III.1 : Les coordonnées 3D d'un point P dans le repère $OXYZ$.

Les méthodes de modélisation font l'objet de nombreuses recherches, et il est impossible de les décrire de manière détaillée dans le cadre de ce mémoire. Les trois méthodes les plus fréquentes consistent en une approximation des volumes ou des surfaces

²On peut, pour faciliter certains calculs, préférer un repère orthonormé [FOLE 90]

par la **géométrie constructive**, par la **modélisation polyédrique**, ou par morceaux de **courbes et surfaces courbes**. Je me limiterai ici à l'étude de la construction de volumes et de surfaces par la géométrie constructive et par le modèle polyédrique. Ce dernier étant le plus simple et le plus répandu vu la simplicité et la rapidité de son principe de génération d'objets 3D. L'étude de l'approximation par courbes et surfaces courbes est complexe, et l'entreprendre en quelques pages serait une gageure. Cependant, pour se rendre compte de l'intérêt de ce type de modélisation, je propose ensuite de la décrire brièvement. Je citerai enfin quelques techniques de modélisation particulières dédiées à certains types d'objets.

III.2.2 La géométrie constructive

La géométrie constructive propose une modélisation de type combinatoire de l'espace tridimensionnel. Elle permet, au moyen d'un vocabulaire fini de formes élémentaires simples et de règles de construction, de définir des objets d'ordre de plus en plus grand à partir d'objets d'ordre inférieur.

A. La structure modulaire

L'idée de base est de considérer que tout objet complexe peut être obtenu par assemblage d'objets simples. Cela implique que l'on s'est fixé une borne inférieure à la profondeur de la description. On a donc postulé l'existence d'atomes indissociables, des briques élémentaires servant de modules à la construction.

Cette structure modulaire présente deux avantages majeurs :

- Le degré de complexité de la description, limitée par la complexité maximale d'un module, peut être choisi arbitrairement en fonction des buts poursuivis et du matériel utilisé. On peut se limiter ainsi, dans certaines applications, à des parallélépipèdes rectangles alors que pour d'autres, on sera amené à considérer des pyramides, des cylindres, etc.

- Les objets de base étant d'un type bien particulier, les algorithmes d'élimination des parties cachées³, par exemple, n'auront nul besoin d'être de généralité maximale mais pourront tenir compte des propriétés particulières de ces objets. Cet avantage fournit un gain en taille et en complexité des programmes d'application, mais également en temps calcul.

³Ces algorithmes seront étudiés dans le chapitre IV.

B. Les opérateurs d'Euler

On peut définir, sur ces modules de base considérés comme l'ensemble des points intérieurs aux volumes limités par leurs surfaces, des opérations ensemblistes au moyen de trois opérateurs :

- L'intersection de deux primitives de base, qui est définie comme étant l'ensemble des points appartenant simultanément aux deux.

- La réunion, qui permet de construire des objets composés à partir d'objets simples.

- La complémentation (ou la différence), qui est définie comme étant l'ensemble des points appartenant à un volume sans appartenir à un autre. Un cube troué pourra ainsi être obtenu en appliquant cet opérateur à un cube et à un cylindre, comme dans la figure III.2 [BRET 88].

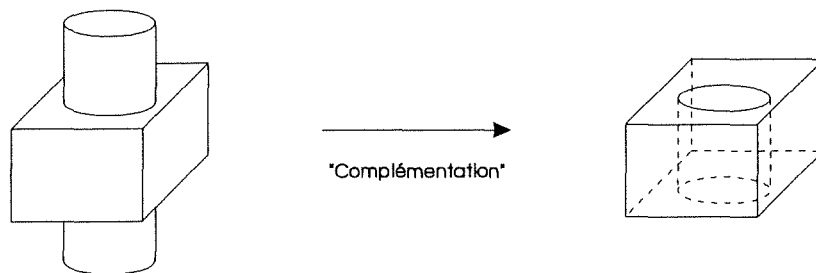


Figure III.2 : L'opérateur de différence appliqué à un cube et à un cylindre.

C. Les domaines d'application

Cette méthode est par exemple utilisée par les chimistes pour modéliser des structures moléculaires ; la primitive de base étant une sphère figurant les atomes et un cylindre matérialisant les liaisons. En mécanique industrielle, où les objets ont souvent des formes géométriques simples et régulières, ce modèle est bien adapté à la manipulation d'assemblage et l'étude de comportement lors de simulations. Les architectes, pour qui la modularité est un langage familier, utilisent également de telles méthodes pour modéliser des structures complexes.

En fait, partout où un langage modulaire et combinatoire peut être employé (modélisation de structures chimiques, architecture intérieure, fabrication de meubles,...) un tel modèle de description trouve des applications évidentes.

III.2.3 Le modèle polyédrique

L'hypothèse de base de la modélisation polyédrique est qu'une surface peut être approchée par une liste finie de polygones dont les sommets appartiennent à la surface. Le nombre de polygones définit la précision de l'approximation polyédrique. On utilise aussi couramment les termes *faces* ou *facettes* pour désigner ces polygones. Chaque facette est contenue dans un plan de l'espace dit *plan support* de la facette. Elle est formée par des segments de droite appelés *arêtes*. Enfin, chaque arête est déterminée et orientée par ses deux extrémités : les *sommets*. Le terme *polyédrique* provient du fait qu'un volume dont la surface est constituée d'un nombre fini de facettes planes est nommé *polyèdre*. Les polyèdres les plus simples sont le cube, les pyramides, les prismes. Un polyèdre plus complexe est montré à la figure III.3.

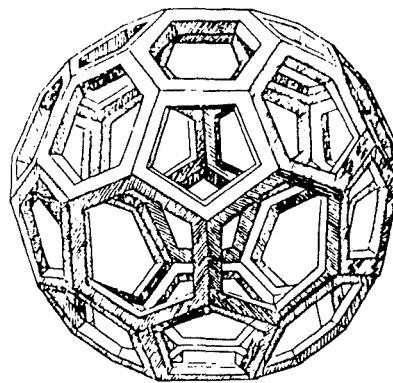


Figure III.3 : Un polyèdre dessiné par Léonardo da Vinci pour le *De divina proportion* de Fra Luca Paccioli. Extrait de [BUSE 88].

A. La description polyédrique

La description d'une surface polyédrique regroupe deux types d'informations :

- Des **informations géométriques** précisant des positions, des distances, et qui sont relatives à des mesures sur les composants de l'objet. Elles concernent donc la liste des triplets de nombres (x,y,z) , correspondant aux coordonnées géométriques des points échantillonnés sur la surface.

- Des **informations topologiques** précisant les connexions entre les primitives de la description. Elles concernent la structure de données associée à la description polyédrique. En effet, la seule liste des sommets est insuffisante pour représenter un volume car rien n'indique comment ils sont reliés pour définir la surface du volume. Il devient alors nécessaire de mémoriser les arêtes, qui relient deux sommets et qui permettent de dessiner le squelette filiforme d'un objet, ainsi que les facettes, qui permettent la détermination des parties cachées.

Plutôt que de donner une représentation explicite, où les coordonnées de sommets sont données plusieurs fois, il est préférable d'utiliser une structure de données plus adéquate. Celle-ci doit comporter la liste des sommets, des arêtes, et des facettes des objets de la scène. Une facette *pointe* vers la liste de ses arêtes, et une arête *pointe* vers ses deux sommets.

Les avantages de cette représentation sont évidents. Une transformation géométrique ne sera appliquée qu'une fois à chaque sommet et surtout, on empêche de la sorte qu'un sommet ne puisse diverger en deux triplets de coordonnées différents suite à l'accumulation d'erreurs d'arrondis, rendant impossible l'identification de toutes les arêtes aboutissant à un sommet unique. Un autre avantage est de pouvoir facilement *modeler* le volume en déplaçant les sommets. Cette représentation d'un volume est intéressante car elle permet assez facilement d'effectuer, par exemple, des *tests de consistance* pour vérifier que chaque facette ait une arête en commun avec une autre.

D'autres informations peuvent également être maintenues. Elles peuvent sembler redondantes, mais peuvent épargner un précieux temps de calcul pour certains algorithmes mis en oeuvre lors de l'affichage. On peut citer par exemple, le *volume englobant* de chaque objet, afin de pouvoir déceler aisément les objets pouvant se masquer, et une information plus utile encore : le *vecteur normal à la facette*.

B. Le vecteur normal

Imaginons un vecteur N perpendiculaire au plan support d'une facette d'un volume et qui donne vers l'extérieur de ce dernier. En définissant le sens du parcours de toutes les arêtes de la facette ; on peut, d'après la règle de la main droite, symboliser le vecteur normal N . Pour ce faire, on pose la main droite sur la facette de sorte que les doigts indiquent le sens du parcours des arêtes de la facette considérée. Le pouce montre alors l'extérieur du volume et symbolise le vecteur normal N , comme indiqué à la figure III.4 extraite de [BUSE 88].

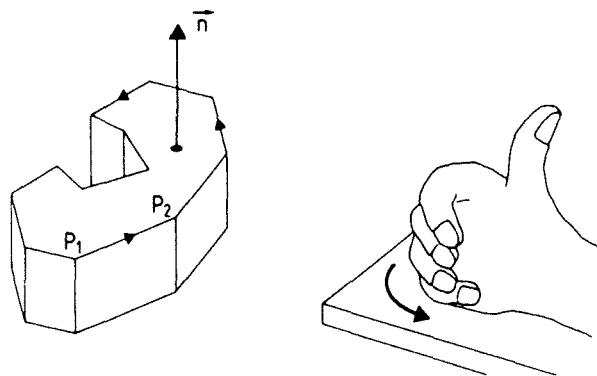


Figure III.4 : La règle de la main droite définit la "normale" à une facette orientée.

III. La synthèse de scènes 3D

L'information nécessaire au calcul du vecteur normal aux facettes est conservée dans la description polyédrique. Les facettes étant des polygones-plans, on peut en effet calculer le vecteur normal N de la manière suivante : soit la facette A , définie par les trois sommets orientés P_1, P_2, P_3 . Son vecteur normal peut être obtenu par le produit vectoriel $P_1P_2 * P_2P_3$.

Cependant, si les trois sommets de la facette sont alignés, le plan de la facette est indéterminé. Il est possible alors de trouver une normale approchée comme la somme des produits vectoriels des couples d'arêtes consécutives de la facette [NEWM 79]. Si l'on suppose, par exemple, que la face est composée des sommets numérotés et orientés de P_1 à P_n ; on peut démontrer que la somme des produits vectoriels

$$M = P_1P_2 * P_1P_3 + P_1P_3 * P_1P_4 + \dots + P_1P_{n-1} * P_1P_n$$

est un vecteur de même direction que N dont la longueur est égale à l'aire de la facette (s'il n'y a pas de polygones intérieurs à la facette). Le vecteur normal N est alors donné par :

$$N = \frac{M}{||M||} \quad \text{où } ||M|| \text{ est la longueur du vecteur } M.$$

Nous verrons par la suite l'intérêt de ce vecteur normal N , qui va permettre d'attribuer des propriétés supplémentaires, comme des couleurs et des intensités, à chaque facette d'un volume.

C. La création d'objets polyédriques

Il est assez aisé de générer des volumes à facettes planes de manière interactive. Deux méthodes se sont imposées : la **révolution** et l'**extrusion**. La génération de la structure de données décrivant ces types de volumes peut être réalisé de manière automatique.

1) La révolution

Il suffit de considérer une ligne polygonale, qui sera appelée la *génératrice*. Par rotation de cette génératrice autour d'un axe (vertical ou horizontal), on génère un volume ou une surface de révolution (si la rotation est inférieure à 360°). La figure III.5 montre la génération d'un verre créé par révolution.

III. La synthèse de scènes 3D

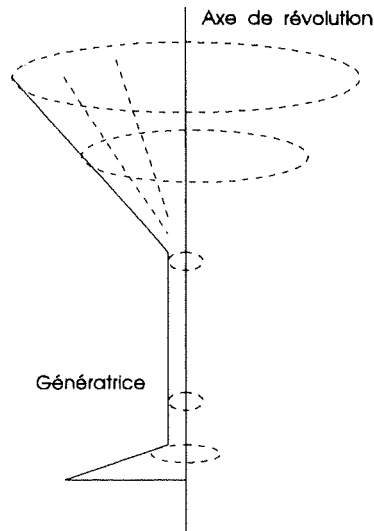


Figure III.5 : Un verre créé par révolution.

2) L'extrusion

Il suffit ici de considérer la base de l'objet à générer. Par *élévation* de cette base suivant un axe d'élévation, il est possible de générer des volumes ou des surfaces (si l'on ne referme pas le dessus de l'objet par une même base). La figure III.6 montre la génération d'une surface par extrusion.

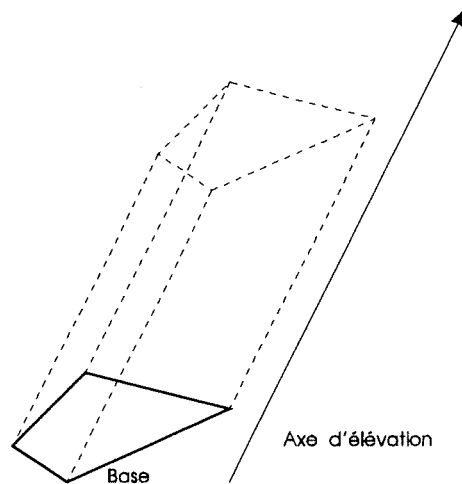


Figure III.6 : Une surface générée par extrusion.

D. La structure hiérarchique

Un grand nombre d'objets, comme pratiquement tous les appareils mécaniques sont constitués de pièces plus simples qui sont connectées par des articulations permettant un certain degré de mobilité. Il est dès lors intéressant, lors de la conception de la structure de données, d'adopter une composition hiérarchique des pièces d'un objet permettant d'obtenir

ainsi des dépendances de mouvements de pièces par rapport à d'autres. Nous verrons comment mettre en oeuvre une telle modélisation hiérarchique dans la partie pratique (Chapitre V) de ce mémoire.

E. Les domaines d'application

La plupart des objets peuvent à présent être modélisés facilement par une approximation polyédrique. On utilise très souvent une modélisation par facettes triangulaires (donc planes).

Les *digitaliseurs* 3D permettent, en pointant sur des positions-clés d'une surface, d'en déterminer les coordonnées spatiales. Il est donc possible avec ce genre d'appareil d'échantillonner des points d'une surface de manière interactive et, dans un deuxième temps, de définir un modèle polyédrique passant par ces points et approchant la surface digitalisée. Des algorithmes générant automatiquement les liaisons polyédriques entre les points digitalisés existent.

Le modèle polyédrique est très utilisé en infographie. Cependant, la représentation sur écran d'une structure polyédrique demande un temps calcul considérable (algorithmes incrémentaux). En effet, à chaque fois que la position de l'observateur est modifiée, il faut recalculer les parties visibles de la structure. On utilise pour cela des algorithmes d'élimination des parties cachées, qui seront traités dans le chapitre IV.

Les méthodes de rendu réaliste destinées à ce type de modélisation font encore l'objet de nombreuses recherches. Mais, par exigence d'esthétisme, on peut affirmer que ce modèle sera délaissé au profit de l'approximation par morceaux de courbes et surfaces courbes.

III.2.4 Les courbes et surfaces courbes

Bien que toute courbe puisse être approchée par des segments de droite jointifs (tel le modèle polyédrique), certaines applications infographiques ne peuvent se satisfaire de cette représentation simpliste. L'utilisation de surfaces courbes apporte des avantages importants pour la modélisation d'objets. En effet, les objets réels sont souvent composés de formes douces, par rapport aux formes anguleuses obtenues à partir de segments de droite.

Une grande variété d'algorithmes est utilisée pour l'affichage de courbes. Seules trois techniques sont couramment utilisées : Les *B-Splines*, les *β -Splines*, et les *courbes de Bézier*. Ces outils mathématiques sont basés sur l'idée de base suivante : une courbe peut être entièrement définie par une suite de points de contrôle. Deux familles de courbes existent : les courbes d'interpolation, qui passent par tous les points de contrôle, et les

courbes d'approximation, qui ne passent pas par ces points, mais à proximité, étant ainsi guidées (par la technique des moindres carrés). Les propriétés de ces courbes conduisent à l'utilisation d'équations paramétriques complexes, présentées dans le détail dans [FOLE 90].

Les *B-Splines*, introduites par Barsky [BARS 83] forment une famille de courbes paramétriques particulièrement intéressantes, qui permettent de modifier localement l'allure de la courbe, en modifiant la position d'un point de contrôle. Les β -*Splines* sont une généralisation des *B-Splines* et permettent de contrôler encore mieux la forme de la courbe obtenue. Les courbes de Bézier [BEZI 87] représentent une approche plus classique, mais pouvant se ramener également aux *B-Splines*. La modification d'un seul point de contrôle d'une courbe de Bézier entraîne une modification de la courbe entière ; à l'instar des *Splines*, le contrôle est global. Toutes ces courbes peuvent être définies aussi bien dans le plan (2D) que dans l'espace (3D), où elles permettent d'engendrer des surfaces (voir figure III.7).

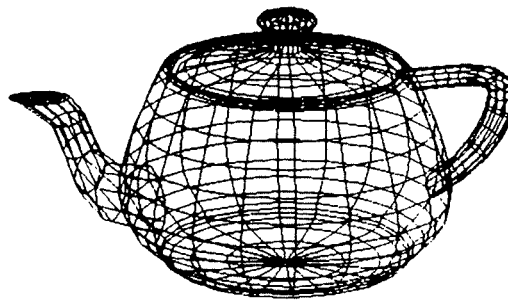


Figure III.7 : Un objet modélisé par 32 éléments de surface de Bézier définis par 306 points de contrôle.

Les surfaces courbes sont à présent indispensables pour la CAO en mécanique, où la génération de volumes fait intervenir de plus en plus des critères esthétiques. Les courbes de Bézier ont d'ailleurs été développées pour le dessin des carrosseries automobiles. De plus, une courbe paramétrique est stockée de manière plus efficace en mémoire, son traçage est plus précis que le modèle polyédrique (pas d'effet de cassure) et une modification de la courbe par ses points de contrôle est aisée. On peut également souligner que le contrôle local est un avantage décisif des *Splines* sur les courbes de Bézier.

III.2.5 Les modèles "évolués"

Les modèles traditionnels (géométrie constructive, modèles polyédriques, surfaces paramétriques,...) dont il vient d'être question s'appliquent à de *bons* objets, c'est-à-dire à des volumes limités par des surfaces lisses et continues. Des difficultés apparaissent dès que l'on a affaire à des surfaces très complexes, plissées, ou floconneuses [BLIN 82], ou s'il

s'agit de fluides turbulents ou de structures en vibration, et plus généralement pour toutes les formes naturelles. D'autres modèles sont dès lors nécessaires. Parmi les plus utilisés, on peut citer :

- Les **objets fractals** [MAND 75] , qui est une méthode très efficace permettant de synthétiser des objets extrêmement irréguliers (terrains, feuilles, nuages, ...). La technique des fractales est reprise dans certains modèles stochastiques [CATM 74].

- Les **systèmes de particules**, qui permettent la modélisation d'objets flous (nuages, feux, fluides turbulents) [REEV 83].

D'autres modèles ont vu le jour, axés sur un type très particulier d'objets à modéliser. On doit, par exemple, la **synthèse de tissus** à Weil [WEIL 86], la **modélisation de vagues** à Peachey [PEAC 86], des **coquillages** à Kawaguchi [KAWA 82], des **structures végétales** d'Oppenheimer [OPPE 86], et bien d'autres ...

III.3 Les transformations et la visualisation 3D

III.3.1 Introduction

La construction du modèle d'un objet dans la mémoire de l'ordinateur constitue une simulation de la réalité et, comme telle, en autorise toutes sortes de manipulations et de transformations.

On peut ainsi :

- modifier l'allure de l'objet en redéfinissant sa structure,
- modifier la position géométrique de l'objet (par translation, rotation,...),
- obtenir différents points de vue de l'objet (se déplacer par rapport à lui, en faire le tour, rentrer dedans,...),
- effectuer des calculs sur certaines propriétés de l'objet (volumes, poids, calculs de résistance, ...),
- entreprendre l'étude de déformations (par la méthode des éléments finis),
- placer le modèle dans certains contextes pour simuler un environnement physique particulier et en observer les effets,
- dialoguer avec le processus de génération par des techniques interactives pour modifier en temps réel des paramètres,...

Les informations géométriques⁴ du modèle seront modifiées par des manipulations géométriques. Les informations topologiques⁴ le seront dans le cas d'une redéfinition de la structure du modèle.

Les **transformations géométriques** jouent un rôle important dans la modélisation et la visualisation d'une scène. Elles permettent en effet de *façonner* une scène, en effectuant des rotations, des translations, des changements d'échelle ou autres manipulations géométriques sur les différents objets qui la constituent.

Il est cependant impossible dans le cadre de ce mémoire, d'expliciter les multiples étapes pour arriver aux formulations mathématiques correspondant à ces transformations. Ceci requiert en effet le rappel de la géométrie en coordonnées cartésiennes et homogènes, ces dernières étant plus appropriées aux techniques infographiques. J'expliciterais cependant pourquoi l'utilisation de la formulation mathématique matricielle en coordonnées homogènes est utile, et je donnerai ensuite quelques exemples de matrices de transformations élémentaires couramment utilisées en infographie. Je présenterai ensuite un type de transformation particulier, sans lequel la visualisation de la scène sur une surface 2D tel l'écran serait impossible : la **projection géométrique**. Mais auparavant, je dirai quelques mots sur deux notions importantes : l'*espace objet* et l'*espace image*.

III.3.2 L'espace objet et l'espace image

Quel que soit le modèle retenu pour représenter un objet de l'espace, celui-ci sera toujours associé à une structure de données faisant référence à un espace, dit **espace objet**. Afin de réaliser la représentation graphique 2D de cette structure de données, un **espace image** est nécessaire.

L'*espace objet* est un espace de description mathématique associé au modèle géométrique dans lequel le point est représenté par un triplet de coordonnées entières ou réelles. On appelle **repère absolu** le système d'axes orthonormés lié à l'*espace objet* et dans lequel sont décrits les volumes à modéliser. Le centre et les directions des axes de ce repère absolu sont théoriquement arbitraires. Cependant, le choix de ce référentiel est généralement fait de façon à faciliter la description des objets modélisés. Ainsi, pour un volume de révolution, par exemple, on aura intérêt à prendre comme axe de révolution un des axes de coordonnées.

⁴Définies précédemment lors de l'étude de la modélisation polyédrique.

Par rapport à l'*espace objet*, l'*espace image* en traduit une représentation graphique 2D et n'en constitue par conséquent qu'une vue partielle, une coupe, ou encore une interprétation (en tant que dépendant de la position de l'observateur et du mode d'éclairage)⁵.

Le passage du premier espace au second se fait, dans le cas tridimensionnel, par une projection géométrique et un calcul de la couleur selon certains paramètres d'éclairage et de visibilité.

III.3.3 Les transformations géométriques

Toute transformation géométrique peut être mathématiquement représentée par des équations linéaires pour chacune des coordonnées d'un point, ou de manière plus élégante par une matrice :

Soit une transformation $T : (x_1, y_1, z_1) \Rightarrow (x_2, y_2, z_2)$.

Elle peut être représentée par le système d'équations :

$$x_2 = a_{11} * x_1 + a_{12} * y_1 + a_{13} * z_1$$

$$y_2 = a_{21} * x_1 + a_{22} * y_1 + a_{23} * z_1$$

$$z_2 = a_{31} * x_1 + a_{32} * y_1 + a_{33} * z_1$$

ou sous forme matricielle :

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} * \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$

En pratique, plusieurs transformations géométriques seront nécessaires pour obtenir l'opération que l'on désire réaliser sur un objet particulier de la scène (par exemple, une rotation, deux translations et un changement d'échelle). Or, l'utilisation de la représentation matricielle en coordonnées cartésiennes ne permet qu'une composition simple de rotations et de changements d'échelle. Les translations ne peuvent être combinées avec ces dernières. C'est pour cette raison que l'on préfère travailler en **coordonnées homogènes**. On peut alors composer tout type de transformation ; la résultante, quelle que soit la complexité de la composition, sera une unique matrice de transformation.

⁵Cet espace image est ainsi représenté par la mémoire d'écran considérée comme structure linéaire de pixels caractérisés par une adresse, ou encore deux coordonnées entières "écran", et une valeur de couleur.

A. Les coordonnées homogènes

Les mathématiciens ont souvent constaté qu'il était souvent plus avantageux de résoudre un problème N-dimensionnel dans un espace à N+1 dimensions. L'idée est donc de décrire un point de l'espace à N dimensions par un vecteur de N+1 coordonnées. La coordonnée supplémentaire est nommée w. Ainsi, dans le cas de l'espace, le passage des coordonnées cartésiennes en coordonnées homogènes peut être obtenu par :

Coordonnées cartésiennes \Rightarrow Coordonnées homogènes

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \begin{pmatrix} x' \\ y' \\ z' \\ w \end{pmatrix}$$

avec $x' = w * x$, $y' = w * y$, et $z' = w * z$.

La valeur w est appelée *facteur d'échelle*. En pratique, on choisit simplement la valeur 1. Le passage inverse, des coordonnées homogènes aux coordonnées cartésiennes, consiste à diviser chacune des coordonnées (x, y, et z) par la dernière (w), sauf si la composante w est nulle. Dans ce dernier cas, les coordonnées homogènes permettent de représenter aisément des points à l'infini. Ces coordonnées homogènes disposent également de propriétés très intéressantes pour les applications infographiques. Comme on l'a déjà mentionné, elles permettent la composition (par multiplication matricielle) de transformations géométriques telles les translations, les rotations, les changements d'échelle,... Enfin, son usage aide à limiter les divisions numériques.

A titre d'illustration de ce dernier avantage [BUSE 88], considérons une transformation homothétique⁶ de facteur 1/3. En coordonnées non homogènes, cette application est formulée par $(x, y, z) \Rightarrow (x/3, y/3, z/3)$. Il faut donc réaliser trois divisions pour chaque point de la structure de données, d'où un temps calcul considérable. En coordonnées homogènes, par contre, il suffit de multiplier la dernière coordonnée par 3, pour obtenir la même transformation ; $(x, y, z, 1) \Rightarrow (x, y, z, 3)$.

⁶Cette transformation permet de réduire d'un facteur 3 les coordonnées d'un objet.

Si l'on reprend la notation matricielle de la transformation T définie précédemment, on obtient en coordonnées homogènes ;

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

B. Quelques exemples de transformations

On mentionne ici quelques exemples de matrices de transformations géométriques élémentaires couramment utilisées en infographie. On entend par transformation élémentaire, des transformations qui ne modifient pas les relations d'angle dans la structure des objets. Elles sont données en coordonnées homogènes.

La translation (Tx, Ty, Tz) :

$$\begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La rotation d'angle A autour de l'axe OX :

$$\begin{pmatrix} \cos A & -\sin A & 0 & 0 \\ \sin A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La rotation d'angle A autour de l'axe OY :

$$\begin{pmatrix} \cos A & 0 & -\sin A & 0 \\ 0 & 1 & 0 & 0 \\ \sin A & 0 & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La rotation d'angle A autour de l'axe OZ :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & -\sin A & 0 \\ 0 & \sin A & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le changement d'échelle de rapport K et de centre l'origine :

$$\begin{pmatrix} K & 0 & 0 & 0 \\ 0 & K & 0 & 0 \\ 0 & 0 & K & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

III.3.4 Les projections géométriques

Le modèle d'un objet dans la mémoire d'un ordinateur regroupe toute l'information sur cet objet, mais est impuissant à le rendre visible. Pour en avoir une perception réaliste, il faut en donner une vision partielle, c'est-à-dire le filtrer, et le projeter de façon à en faire un échantillonnage plausible du monde par un oeil hypothétique [BRET 88]. Un type particulier de transformation est donc nécessaire : les projections géométriques. Cette transformation exprime le passage de l'espace au plan, de l'univers 3D à l'univers 2D. Pour ce faire, toute une série de paramètres seront indispensables, comme par exemple la définition d'un volume de vision qui spécifie ce que l'observateur *voit* de la scène. Ce volume de vision sera utile aux algorithmes de découpage, puisque seules les parties intérieures à ce volume doivent apparaître.

Il existe différents types de projections. Les projections parallèles suivent les règles de la géométrie affine classique, alors que les projections perspectives possèdent leur propre géométrie (différente de la géométrie euclidienne). En perspective, deux lignes parallèles ne le sont plus après projection (sauf si elles étaient parallèles au plan de projection). La longueur d'un segment est réduite par un facteur proportionnel à sa distance par rapport à l'observateur. Par conséquent, aucune information métrique ne peut être extraite d'une mise en perspective d'une scène. Par contre, la perspective produit une meilleure impression de profondeur que les autres, car elle correspond à notre vision naturelle.

Pour les mêmes raisons que précédemment, il n'est pas possible dans le cadre de ce mémoire d'aborder l'étude systématique des transformations de projection. Tous les détails concernant celles-ci peuvent être trouvés notamment dans [FOLE 90].

III.4 Conclusion

Il est important de souligner que la synthèse proprement dite (l'affichage sur écran) de la structure de données décrivant une scène est basée sur les mêmes calculs pour ce qui est de la modélisation, des transformations et des projections. En effet, une fois que les sommets, arêtes, facettes et la couleur de ceux-ci sont construits et stockés, la production d'une image se fait en *pipeline* dont le schéma simplifié est présenté à la figure III.8.

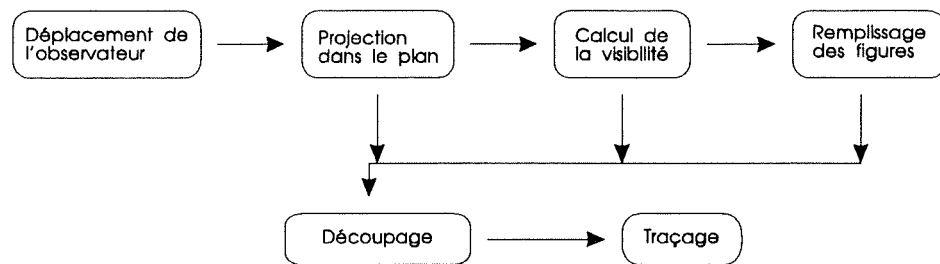


Figure III.8 : Production en pipeline d'une image.

Il est à noter que les principaux algorithmes de visualisation (opérations de remplissage des figures, de découpage, et de traçage) font généralement partie intégrante de la bibliothèque graphique *hardware* minimale de la plupart des stations de travail professionnelles.

La structure de données affichée telle qu'elle est modélisée dans la mémoire de l'ordinateur (par la liste des faces, des arêtes, et des sommets des objets de la scène) ne permet qu'une vision très schématique de la réalité. Cette représentation graphique obtenue par l'affichage de ces points et arêtes remplace la surface qu'ils symbolisent. Le terme consacré est celui de *modèle filaire* ou "*fil-de-fer*" qui traduit, de façon imagée, l'apparence ainsi obtenue. Ce type de représentation, s'il peut suffire pour certaines applications bien spécifiques, ne saurait répondre aux exigences actuelles en matière de réalisme dans le domaine de l'imagerie de synthèse.

Après avoir abordé le problème de modélisation et de visualisation des scènes, je vais à présent me concentrer sur cet aspect très attrayant de la synthèse d'images, et sur lequel le sujet de ce mémoire s'est focalisé : le rendu réaliste des scènes 3D.

IV. LE RENDU REALISTE DE SCENES 3D

IV.1 Introduction

L'*image réaliste* est un exemple d'image synthétique utilisant la structure d'image tridimensionnelle comme squelette et en lui adjoignant un certain nombre d'autres modèles destinés à simuler l'apparence des objets réels. Des recherches poussées ont été menées et une très importante littérature a été publiée sur le sujet, accompagnée d'exemples photographiques d'un étonnant réalisme : ACM SIGGRAPH, IEEE Computer Graphics and Applications, ACM Transactions On Graphics, CAD/CAM, Computer-Aided Design, Computer Graphics, Computer Graphics and Image Processing, Computer Graphics for Management, The Computer Graphics Software News, Computer Graphics World, Computer and Graphics, Display Technology and Applications, NCGA Computer Graphics News, etc...

Le rendu réaliste d'une scène 3D est réalisé de manière incrémentale. Examinons-en les étapes...

IV.2 Les étapes du rendu réaliste

La première amélioration pour rendre crédible une scène tridimensionnelle, est de résoudre le *problème des parties cachées*. En effet, nous ne comprenons notre environnement, et en particulier nous ne sommes sensibles à la notion de profondeur, que parce que l'information qui nous en parvient est partielle : tel objet sera vu derrière tel autre, justement parce qu'il est en partie caché.

Une deuxième amélioration à apporter à la scène 3D est de la munir d'un *modèle d'illumination* (ou d'éclairage). Nous ne voyons en effet une scène réelle que parce qu'elle est éclairée. De plus, pour les mêmes raisons que celles évoquées ci-dessus à propos de la compréhension de la profondeur, la sensation de relief provient de la non uniformité de cet éclairage. Les modèles élaborés à ce jour consistent tous à approcher, par des formules inspirées de l'optique géométrique, les lois physiques gouvernant la propagation des ondes lumineuses dans les milieux transparents, translucides ou opaques. Ces modèles d'illumination doivent donc prendre en compte le rendu de la *transparence* et des *ombres*.

Un troisième aspect concernant le réalisme d'une image 3D est lié à la microstructure des surfaces : leur *texture*. Les objets naturels sont en effet rarement d'un type géométrique simple et leur approximation par des surfaces continues, lisses et régulières n'est qu'une schématisation simplificatrice. Prenons l'exemple d'une boule de billard et d'une orange... Ces *objets* répondent bien, en première approximation, à la même

description sphérique. Cependant, ils révèlent, à une observation plus attentive, des différences d'aspect superficiel notables ; alors que la boule de billard est très proche de son modèle mathématique, l'orange s'en éloigne d'autant plus que l'échelle d'observation est plus petite.

Etudions maintenant plus en détail ces étapes du rendu réaliste d'une scène synthétique. Je présenterai ensuite un algorithme très apprécié en infographie vu la simplicité de son principe et son excellent résultat : le *lancer de rayons*. Je terminerai en décrivant simplement une méthode récente et très efficace, mais cependant trop complexe pour l'aborder dans ce mémoire : la technique de *radiosité*.

IV.3 L'élimination des parties cachées

IV.3.1 Introduction

Grâce à une vision partielle d'une scène, on peut appréhender la notion de profondeur. La projection perspective d'une scène, en modèle fil-de-fer, constitue par conséquent une première approximation de la réalité. Mais l'élimination des parties cachées de cette scène est nécessaire pour obtenir un rendu réaliste. De nombreuses recherches ont été effectuées à ce sujet, et un grand nombre d'algorithmes ont été proposés. Ces algorithmes acceptent comme paramètres d'entrée la structure de données 3D des objets d'une scène, la position de l'observateur, et d'autres paramètres de visualisation. Ils produisent en sortie l'aspect (2D) qu'aurait la scène pour cet observateur dans la réalité (voir figure IV.1).

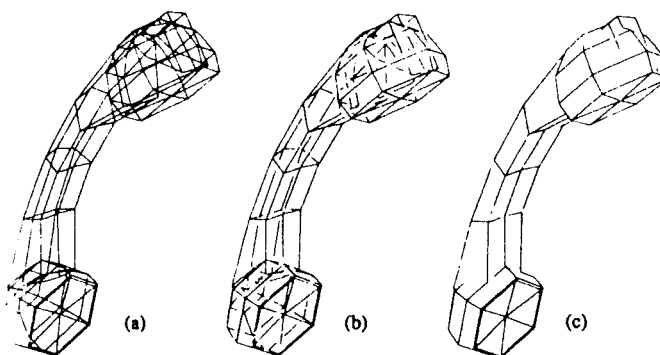


Figure IV.1 : Polyèdre approchant une surface, a) et b) en fil-de-fer, c) avec élimination des parties cachées.

Cet aspect de l'image réaliste est maintenant maîtrisé et une importante littérature a été publiée sur le sujet.

IV.3.2 Classification des algorithmes

On peut distinguer trois types d'algorithmes d'élimination des parties cachées [SUTH 74] :

- Ceux qui opèrent dans l'*espace objet*⁷ et qui sont plus spécialement orientés vers l'élimination des arêtes cachées, comme l'algorithme de Roberts [ROBE 64].
- Ceux qui opèrent dans l'*espace image* et qui sont employés pour colorier les surfaces en calculant également la couleur en chaque point, comme l'algorithme de l'horizon mobile [ROGE 85] ou l'algorithme du lancer de rayons [KAY 79].
- Ceux qui utilisent des listes de priorité et qui travaillent dans les deux espaces en même temps. On retrouve dans cette catégorie les algorithmes de Schumacker [SCHU 69], de Newell-Newell-Sancha [NEWE 72], de Warnock [WARN 68,69], les algorithmes *Scan-Line*, et l'algorithme de Z-Buffer.

Il est impossible de passer en revue tous les types d'algorithmes d'élimination des parties cachées. Je me limiterai aussi à résumer les deux algorithmes les plus utilisés ; l'algorithme de Z-Buffer et l'algorithme du lancer de rayons.

IV.3.3 L'algorithme de Z-Buffer

Cet algorithme travaille simultanément dans l'*espace image* et dans l'*espace objet*. Son principe est assez simple... Il consiste à n'écrire un pixel⁸ dans la mémoire d'écran que si la distance de l'oeil de l'observateur au point 3D dont ce pixel est la projection est inférieure à toutes les distances de cet oeil aux autres points de l'espace ayant la même projection. Cela suppose de mémoriser toutes les *profondeurs* des points de la scène. Mais puisque seule une comparaison est réalisée, il suffit de stocker dans une mémoire auxiliaire le minimum des profondeurs calculées jusqu'alors. Cette mémoire auxiliaire fournira donc, pour chaque pixel, la profondeur (la coordonnée z) du point de l'espace lui correspondant et qui est le plus proche de l'oeil de l'observateur. Cette mémoire doit être de la même taille que la mémoire d'écran, et est appelée le "Z-Buffer".

⁷Les espaces objet et image ont été définis précédemment, lors de l'étude de la modélisation 3D.

⁸Un pixel (de l'anglais "picture element") représente le plus petit élément adressable d'une image sur un écran. Il correspond à un mot de la mémoire de l'écran..

IV. Le rendu réaliste de scènes 3D

L'algorithme de Z-Buffer peut être décrit en pseudo-code de la manière suivante :

Initialiser le Z-Buffer à une valeur très grande.

Pour chaque facette de la scène,

Faire

 Pour chaque pixel intérieur au contour de la projection de cette facette,

 Faire

 Calculer la profondeur Z du point projeté

 Si Z est inférieur à la valeur stockée dans le Z-Buffer, alors
 remplacer par Z et écrire la couleur du pixel dans la mémoire
 d'écran.

 Fait

Fait

La méthode du Z-Buffer conduit à une consommation mémoire importante. Il faut en effet réserver un Z-Buffer de la taille de la mémoire d'écran, parfois déjà conséquente. De plus, si l'on désire une grande précision, il faut déclarer des zones mémoires de 16 ou 32 bits pour chaque pixel de l'écran.

D'autre part, il est difficile d'obtenir par cette technique des effets de transparence, vu que les facettes sont traitées indépendamment de leurs positions relatives.

IV.3.4 L'algorithme du lancer de rayons

On tente de simuler par cette technique les lois de l'optique géométrique. Celles-ci seront étudiées dans la partie suivante traitant des modèles d'illumination.

Les infinités de rayons issus des sources lumineuses d'une scène et qui, après de multiples réflexions et réfractions sur les objets constituant la scène, arrivent à l'oeil de l'observateur sont responsables de la perception visuelle de celui-ci. En appliquant le principe du retour inverse de la lumière, la méthode du lancer de rayons fait partir au départ de l'oeil de l'observateur des rayons... On remonte alors l'histoire de toutes les intersections de ces rayons à travers la scène.

J'étudierai par la suite plus en détail ce puissant algorithme, tant il est important et efficace pour le rendu réaliste. Mais, on peut déjà donner, en pseudo-code, un résumé de cet algorithme.

Pour chaque pixel P de l'écran,

Faire

Pour chaque objet X de la scène,

Faire

Calculer les intersections de la droite joignant l'oeil de l'observateur au pixel P avec l'objet X.

Fait

Trier ces intersections selon la coordonnée Z.

Si la liste est vide, alors colorier le pixel P dans la couleur du fond, sinon dans la couleur de l'objet correspondant au plus proche.

Fait.

On peut remarquer qu'il s'apparente fortement avec l'algorithme de "Z-Buffer" ; la différence (considérable) se situant dans le calcul de la couleur des pixels de l'image. Le lancer de rayon permet en effet de prendre en compte tous les phénomènes d'illumination, d'ombrage, de réflexion, et de réfraction.

La seule élimination des parties cachées ne justifie pas la mise en oeuvre d'un algorithme aussi coûteux. On préférera généralement d'autres méthodes plus simples. Mais cependant, vu la qualité du résultat obtenu par cet algorithme, il est souvent mis en oeuvre pour réaliser des images de synthèse ou un film d'animation nécessitant un rendu réaliste de très haute qualité.

IV.3.5 Conclusion

L'élimination des parties cachées est la première étape dans l'approche réaliste d'une scène synthétique. Cette opération est souvent réalisée au niveau du matériel. Les stations graphiques professionnelles actuelles proposent toutes un "Z-Buffer" pour réaliser cette opération de manière optimale.

Je vais à présent entamer l'étude de principes qui sont à la base de nombreuses techniques mises en oeuvre dans tout rendu réaliste de qualité : les **modèles d'illumination**.

IV.4 Les modèles d'illumination

IV.4.1 Introduction

Dans les algorithmes d'élimination de parties cachées, on détermine quel est le pixel à colorier. Mais de quelle couleur ? La réponse n'est pas évidente. Si on doit colorier une sphère de couleur verte, une coloration uniforme ne lui donnera pas un aspect réaliste. Ce qui donne l'impression de volume, c'est justement une répartition judicieuse des intensités de couleur sur la surface.

De nombreuses recherches ont été menées concernant la modélisation de scènes tridimensionnelles éclairées. Pour cela, il est nécessaire de synthétiser la lumière...

IV.4.2 La lumière synthétique

Synthétiser la lumière nécessite de prendre en compte un certain nombre de facteurs : la nature des sources lumineuses et des objets éclairés et, bien évidemment, les lois de la propagation de la lumière.

A. La nature des sources lumineuses

Généralement, trois types de sources lumineuses sont retenus :

- La **lumière ambiante** ; qui éclaire toute la scène uniformément.
- Les **sources directionnelles** ; supposées à l'infini, et qui éclairent la scène avec des rayons parallèles à une direction donnée.
- Les **sources ponctuelles** ; supposées placées en un point précis de la scène, et qui rayonnent la lumière de manière omnidirectionnelle (c'est-à-dire dans toutes les directions).

On utilise également un type particulier de source ponctuelle et qui simule l'effet d'un projecteur. Ceux-ci sont contrôlés par leur position, leur direction, et un facteur de concentration spécifiant un *cône de lumière*.

B. La nature des objets éclairés

Les objets éclairés peuvent être de trois types :

- Les **surfaces réfléchissantes** (en partie ou totalement) ; caractérisées par un coefficient qui dépend du matériau qui compose la surface (couleur, rugosité, texture,...).

- Les **surfaces translucides ou transparentes** ; qui permettent la réfraction et la transmission (plus ou moins complète) de la lumière.
- Les **objets mous** ; généralement modélisés par des systèmes de particules ou par des techniques fractales.

C. Les lois de la propagation lumineuse

Les rayons lumineux provenant des sources de lumière changent de direction au contact de surfaces selon deux lois principales :

- les lois de la réflexion ; diffuse et spéculaire et
- les lois de la réfraction ; lors d'un changement de milieu (utile pour la transparence).

Une partie de l'énergie d'un rayon lumineux incident à une surface est absorbée par celle-ci lors du changement de direction. Si la lumière incidente à la surface est de couleur blanche, la partie absorbée par la surface pour chaque composante spectrale (le rouge, le vert et le bleu)⁹ détermine la couleur de cette surface. L'autre partie de l'énergie du rayon incident est soit transmise, soit réfléchi (voir figure IV.2).

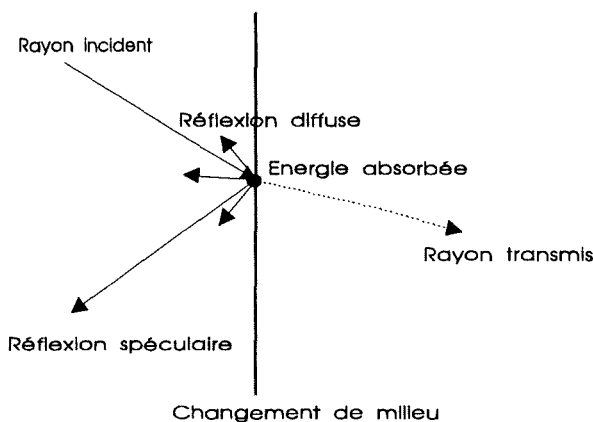


Figure IV.2 : La propagation d'un rayon lumineux incident à une surface.

L'énergie réfléchi par la surface provient de deux contributions :

- la **réflexion spéculaire**, propre à un miroir, qui renvoie le rayon selon une direction précise.

⁹La couleur d'un objet est déterminée par la faculté d'absorption de sa surface des trois couleurs primaires Rouge, Vert et Bleu.

-la **réflexion diffuse**, propre aux surfaces mates, qui correspond à l'absorption puis à la réémission omnidirectionnelle (diffuse).

Les objets métalliques ont une réflexion spéculaire importante, leur conférant cet aspect brillant caractéristique.

Il existe plusieurs modèles d'illumination... J'étudierai ci-dessous celui qui est généralement le plus utilisé : le *modèle de Phong et Blinn* [PHON 73,75], [BLIN 76,77]. Je présenterai ensuite brièvement deux autres modèles ; le *modèle de Cook* [COOK 81,82] et le *modèle de Torrance et Sparrow* [TORR 67].

IV.4.3 Le modèle de Phong et Blinn

Pour expliquer ce modèle, j'adopterai une étude séquentielle des différents éléments le composant, du plus élémentaire au plus complexe...

A. L'illumination diffuse

Le cas le plus simple d'illumination est celui résultant d'une illumination diffuse. La lumière provient de toutes les directions avec une intensité constante en tout point de l'espace, et ne dépend que de la nature des objets éclairés à l'exclusion de toute source lumineuse localisée (ponctuelle ou directionnelle). On parlera alors de lumière ambiante ou de lumière diffuse.

Pour une intensité lumineuse I_a de lumière ambiante qui est reçue en un point, et un coefficient de réflexion de la lumière ambiante K_a , l'intensité lumineuse réfléchie I_{Ambiante} en (renvoyée par) ce point est :

$$I_{\text{Ambiante}} = K_a * I_a$$

Le coefficient de réflexion $K_a = \frac{I_{\text{Ambiante}}}{I_a}$ caractérise la *réflectance* de la surface en ce point. Il est compris entre 0 et 1, et dépend du matériau de la surface. Un objet noir est caractérisé par un coefficient de réflexion valant 0, car il absorbe toute la lumière reçue. A l'opposé un objet blanc est caractérisé par un coefficient valant 1. Un tel objet n'apparaîtra blanc que s'il est éclairé par une source de lumière blanche, alors que pour une source de lumière colorée, il sera perçu de la couleur de la lumière.

Dans le cas de production d'images en couleurs, on travaille en décomposant la couleur en ses trois composantes primaires, appelé *modèle additif*, le Rouge, Vert et Bleu

(d'où le terme anglais RGB), ou selon les trois couleurs complémentaires Cyan, Magenta, et Jaune, appelé *modèle soustractif*. L'équation ci-dessus peut alors s'écrire sous sa forme développée (en synthèse additive) :

Pour la composante Rouge : $I_{\text{Ambiante},r} = K_{a,r} * I_{a,r}$

Pour la composante Verte : $I_{\text{Ambiante},v} = K_{a,v} * I_{a,v}$

Pour la composante Bleue : $I_{\text{Ambiante},b} = K_{a,b} * I_{a,b}$

Il est possible de démontrer [FOLE 90] que la luminosité d'une facette d'une surface éclairée par la lumière ambiante est indépendante de son orientation par rapport à la position de l'observateur de la scène.

B. Critiques

Une telle composante ambiante est insuffisante pour rendre l'aspect réaliste d'une scène tridimensionnelle. Une lumière purement diffuse estompe l'effet de profondeur. Ce type d'illumination ne fait pas ressortir les formes, car pour un objet de couleur unie toutes les facettes auront la même intensité : une sphère sera perçue comme un disque. De plus, on ne peut générer ni ombres, ni reflets.

L'utilité de la lumière ambiante est en fait d'adoucir la lumière trop crue que produirait une unique source lumineuse. Cette technique est employée en photographie, où l'on ne dirige pas un flash directement sur le sujet, mais sur un réflecteur mat pour obtenir une *ambiance*. Par ce biais, les parties non exposées aux sources lumineuses ne seront pas totalement noires.

En pratique, l'éclairement d'une scène est rarement uniforme et provient plutôt de sources lumineuses qui illuminent la scène selon des angles variés, et qui produiront une illumination différente sur chaque facette en fonction de son inclinaison. Apparaîtront également des zones d'ombres qui sont en partie responsables de la perception du relief.

C. La réflexion diffuse

En plus d'une composante ambiante, une scène est toujours illuminée par une ou plusieurs sources de lumière ponctuelle. Toute surface exposée à la lumière de ces sources la réfléchit de manière diffuse. Si la lumière provenant d'une de ces sources arrive perpendiculairement à une surface, pour un même flux lumineux, la quantité d'énergie reçue par unité de surface sera plus grande pour des rayons perpendiculaires que rasants...

D. La loi de Lambert

Plus précisément, la loi de Lambert spécifie que l'intensité lumineuse diffuse réfléchie en un point d'une surface dépend de l'angle α formé entre le rayon incident de la source lumineuse et le vecteur normal à cette surface selon une distribution en Cosinus. L'intensité diffuse réfléchie représente, pour la source ponctuelle d'indice i , :

$$I_{\text{Diffuse},i} = K_d * I_i * \cos \alpha_i \quad ; \text{ avec } 0 \leq \alpha_i \leq \frac{\pi}{2}$$

K_d désigne la constante de réflexion diffuse, qui est fonction du matériau de la surface. Il est compris entre 0 et 1. I_i représente l'intensité lumineuse incidente pour la source ponctuelle i .

L'illumination varie d'une valeur maximale pour $\alpha=0$ degré (incidence normale, source située à l'aplomb de la surface), à une valeur nulle pour $\alpha=90$ degrés (éclairage rasant, source étant située dans le plan tangent à la surface). Pour un angle $\alpha > 90$ degrés, la source étant de l'autre côté du plan tangent, le point n'est plus éclairé et se trouve dans l'ombre. Ces différentes situations sont schématisées à la figure IV.3.

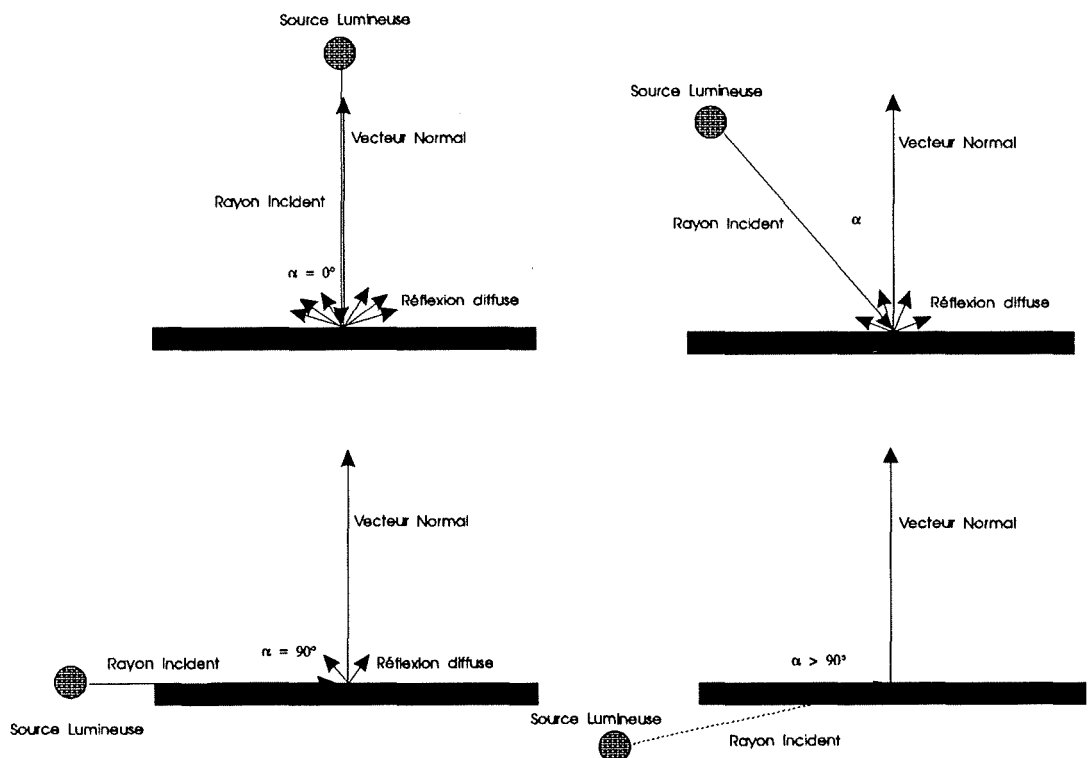


Figure IV.3 : La variation de la réflexion diffuse en fonction de l'angle d'incidence.

IV. Le rendu réaliste de scènes 3D

Comme précédemment pour la lumière ambiante, la position de l'observateur n'intervient pas ; la lumière réfléchie l'est de manière diffuse avec la même intensité dans toutes les directions.

Un premier modèle tenant compte des effets d'ambiance et diffus (si l'on considère les n sources de la scène) est donc :

$$\begin{aligned} I_{\text{Totale}} &= I_{\text{Ambiante}} + I_{\text{Diffuse}} \\ &= I_a * K_a + \sum_{i=1}^n K_d * I_i * \cos \alpha_i \end{aligned}$$

E. Le calcul du Cosinus α

Pratiquement, le calcul du Cosinus α peut se faire par le produit scalaire :

$$\cos \alpha = \frac{L_i * N}{||L_i|| * ||N||}$$

L_i représente le vecteur dirigé vers la source lumineuse i et N , le vecteur normal à la surface illuminée.

En faisant l'hypothèse que les vecteurs L_i et N sont normés, l'intensité réfléchie peut se calculer par l'expression :

$$I_{\text{Diffuse}} = K_d * I_i * (L_i * N) \quad ; \quad \text{si } ||N|| = ||L_i|| = 1.$$

F. L'atténuation selon la distance de l'observateur

Si deux objets sont placés à même distance de l'observateur, mais à des distances différentes de la source lumineuse, aucune différence ne sera perceptible entre eux en appliquant ce modèle. Or, en réalité, on perçoit lequel est le plus proche de la source en raison de leurs illuminations différentes. En conséquence, le modèle doit être complété en introduisant la notion de distance à une source lumineuse qui va agir en diminution de l'intensité incidente (et donc aussi réfléchie). Selon les lois physiques de la propagation lumineuse, l'intensité lumineuse incidente diminue avec le carré de la distance à la source :

$$I_{\text{Diffuse},i} = K_d * I_i * \frac{1}{D_i^2} * (L_i * N)$$

La distance d'un point de la surface illuminée à la source lumineuse i est symbolisée par D .

IV. Le rendu réaliste de scènes 3D

Cette formule ne peut être utilisée pour une source placée à l'infini, car sa contribution serait alors nulle. De plus, on a remarqué que si la source est placée à distance finie, l'image calculée selon cette formule conduit à une image peu agréable car trop violemment contrastée.

En pratique, on considère une diminution de l'intensité proportionnelle à l'éloignement de la source, donnant des résultats satisfaisants :

$$I_{\text{Diffuse},i} = K_d * I_i * \frac{1}{D_i + M} * (L_i * N)$$

La constante M sert à éviter l'augmentation vers l'infini pour des points très proches de la source d'indice i.

L'intensité réfléchie totale en un point de la surface (s'il existe n sources dans la scène) devient :

$$\begin{aligned} I_{\text{Totale}} &= I_{\text{Ambiante}} + I_{\text{Diffuse}} \\ &= K_a * I_a + \sum_{i=1}^n [K_d * I_i * \frac{1}{D_i + M} * (L_i * N)] \end{aligned}$$

Pour l'obtention d'images couleurs, on utilisera le modèle additif ou soustractif, soit pour ce dernier :

$$\text{Cyan : } I_{\text{Totale},c} = K_{a,c} * I_{a,c} + \sum_{i=1}^n [K_{d,c} * I_{i,c} * \frac{1}{D_i + M} * (L_i * N)]$$

$$\text{Magenta : } I_{\text{Totale},m} = K_{a,m} * I_{a,m} + \sum_{i=1}^n [K_{d,m} * I_{i,m} * \frac{1}{D_i + M} * (L_i * N)]$$

$$\text{Jaune : } I_{\text{Totale},j} = K_{a,j} * I_{a,j} + \sum_{i=1}^n [K_{d,m} * I_{i,m} * \frac{1}{D_i + M} * (L_i * N)]$$

G. La réflexion spéculaire

L'utilisation de la réflexion spéculaire va permettre de corriger la réflexion diffuse qui n'est, en général, qu'une approximation de la réalité. En effet, pour la plupart des matériaux, la lumière réfléchie en un point de la surface ne l'est pas uniformément dans toutes les directions. Il faut tenir compte également de la position de l'observateur.

A l'opposé des surfaces mates, les surfaces brillantes produisent des *reflets* ("highlights") dont la direction dépend de l'incidence de la lumière. Le reflet ne sera visible que si l'observateur se trouve dans une zone restreinte autour de la direction du reflet, et diminuera très rapidement d'amplitude dès qu'il s'en écarte. A l'opposé de la réflexion diffuse, la réflexion spéculaire produit un pinceau de lumière réfléchi très fin et donc directionnel. Contrairement au cas diffus, la lumière réfléchi de manière spéculaire, appelée reflet ou éclat, est de la couleur de la lumière incidente et ne dépend aucunement de celle de l'objet.

H. La réflexion parfaite : La loi de Fresnel

Dans le cas d'un miroir parfait, le rayon lumineux est réfléchi, selon la loi de Fresnel, avec le même angle α , par rapport au vecteur normal à la surface, que le rayon incident mais dans la direction opposée, selon le schéma de la figure IV.4. Le reflet n'est alors perçu par l'observateur que s'il se trouve exactement sur cette droite.

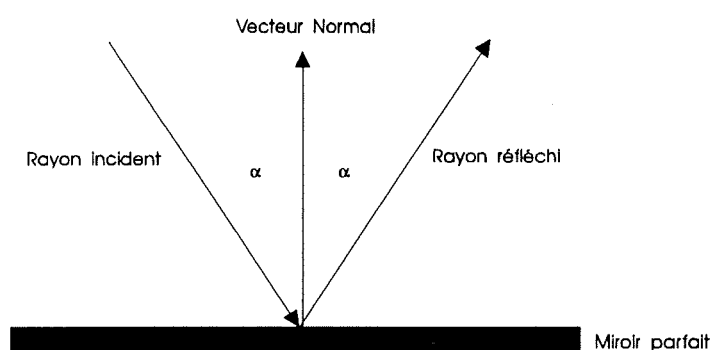


Figure IV.4 : La réflexion spéculaire parfaite.

Si la surface illuminée est très polie, on pourra aussi y percevoir la réflexion d'autres objets comme dans un miroir... Le rendu d'objets vus par réflexion sur une surface n'est simple que par la méthode du *lancer de rayons*, étudiée dans la suite de ce chapitre. Les autres techniques se restreignent à la représentation des éclats.

I. La réflexion imparfaite

Pour des matériaux brillants et non parfaitement réfléchissants, la réflexion spéculaire n'est pas concentrée en un rayon, mais diverge en un faisceau (comme modélisé à la figure IV.5). Si la surface est très polie, très brillante (comme l'acier ou le verre), le pinceau sera très fin alors qu'il s'élargira plus rapidement pour une surface moins proche d'un miroir parfait (bronze, plastique).

IV. Le rendu réaliste de scènes 3D

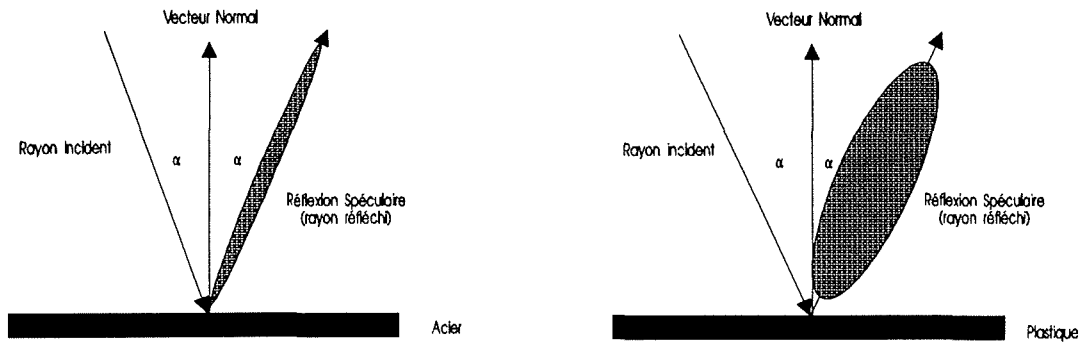


Figure IV.5 : La réflexion spéculaire dans le cas fréquent de miroir imparfait.

J. L'intensité relative à la réflexion spéculaire

A l'intérieur du faisceau réfléchi de manière spéculaire, l'intensité lumineuse diminue très rapidement dès qu'on s'écarte de la direction théorique.

Cette réflexion spéculaire est maximale dans la direction symétrique R du rayon incident I et diminue à mesure que l'angle β entre le vecteur dirigé vers l'observateur O et R augmente (voir figure IV.6). Ce type de réflexion est appelé, pour cette raison *éclairage spéculaire*.

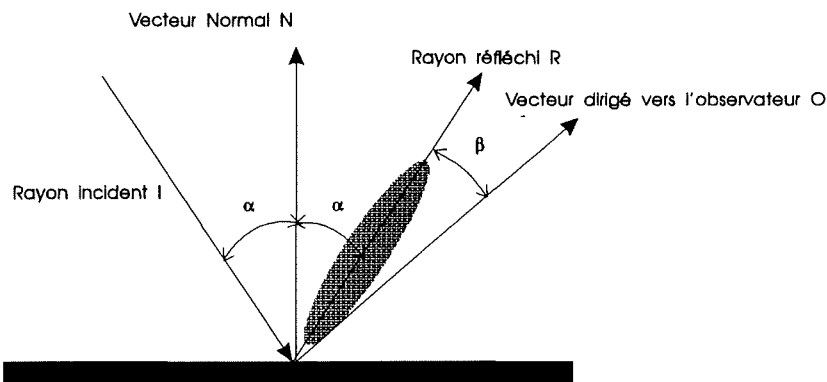


Figure IV.6 : La dépendance de la position de l'observateur et de l'intensité de la réflexion spéculaire.

L'intensité de ce faisceau réfléchi peut être modélisé par une formule tenant compte de ce fait [NEWM 79] , où l'intensité spéculaire réfléchie par une surface S illuminée par une source lumineuse d'indice i est modélisée par :

$$I_{\text{Spéculaire},i} = I_i * W_i(\alpha,S) * (\cos \beta_i)^n$$

I_i spécifie l'intensité de la lumière incidente de la source ponctuelle i.

$W_i(\alpha, S)$ représente le coefficient de *réflectance*, et est fonction de l'angle α d'incidence du rayon issu de la source d'indice i , et caractérise la nature de la surface S . Il est maximum pour $\alpha=90$ degrés, correspondant alors à un éclairage rasant. La figure IV.7 donne quelques exemples de cette fonction pour divers matériaux [NEWM 79]. En pratique, on se contente d'une valeur constante K_S .

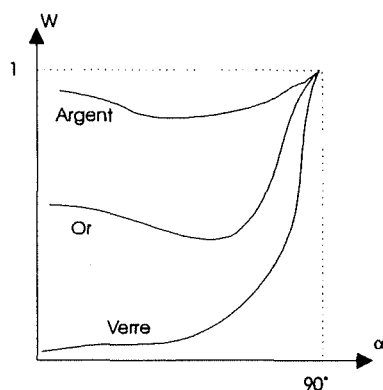


Figure IV.7 : Les valeurs de W pour quelques matériaux en fonction de l'angle d'incidence α .

Pour rappel, β symbolise l'angle entre le rayon réfléchi R et la direction de vision O . Le terme en Cosinus β élevé à la puissance n exprime que la composante spéculaire est maximale si l'oeil est situé sur le rayon réfléchi R ($\beta=0$).

Le coefficient n (compris entre 0 et 1) mesure la brillance, c'est-à-dire la vitesse de variation de l'éclairement en fonction de l'angle β . La valeur de cet exposant dépend du matériau de la surface : de grandes valeurs de n correspondant au comportement des surfaces métalliques, et de petites valeurs correspondant à des surfaces non-métalliques (comme le papier par exemple). Il est compris entre 1 et 200 et vaut l'infini pour un parfait réflecteur. Les valeurs de ce coefficient, qui permet de contrôler le *reflet spéculaire*, sont choisies de manière empirique.

K. L'atténuation de la réflexion spéculaire

Théoriquement, l'éclairement en un point d'une surface décroît à mesure que la distance M entre la source lumineuse et ce point augmente. Cependant, si la source est loin par rapport aux dimensions de la scène, on peut, en première approximation négliger cette variation (c'est le cas pour une source à l'infini, comme le soleil). Les remarques faites précédemment sur l'atténuation de la réflexion diffuse en fonction de la distance restent donc valables ici.

IV.4.4 Le modèle d'illumination simplifiée

En composant les termes dus à la lumière diffuse et aux réflexions provenant des différentes sources lumineuses, le **modèle d'illumination simplifiée** tient compte des effets ambiants, diffus et spéculaires. Il devient (en gardant les termes en Cosinus) :

$$I_{\text{Totale}} = I_{\text{Ambiante}} + I_{\text{Diffuse}} + I_{\text{Spéculaire}}$$

$$= I_a * K_a + \sum_{i=1}^n [(K_d * \cos \alpha_i + K_s * (\cos \beta_i)^n) * I_i * \frac{1}{D_i + M}]$$

A. Le calcul du cosinus β

Pour évaluer cette expression, il reste à trouver une méthode simple de calcul du Cosinus de l'angle β , par exemple en procédant de manière similaire que précédemment pour $\cos \alpha$.

Soit R et O, respectivement les vecteurs unités dans la direction du rayon réfléchi et dans la direction de vision (vers l'oeil de l'observateur). Le Cosinus de l'angle β est égal à leur produit scalaire :

$$\cos \beta = R \cdot O$$

B. Le calcul du vecteur R

On a vu auparavant que la loi de Fresnel postule l'égalité des angles entre le vecteur normal N à la surface et le rayon incident I ainsi que cette même normale et le rayon réfléchi R. Ces trois vecteurs étant dans un même plan, le calcul de la direction du rayon réfléchi R est assez simple. On peut se faire une idée de la solution par construction géométrique (voir figure IV.8).

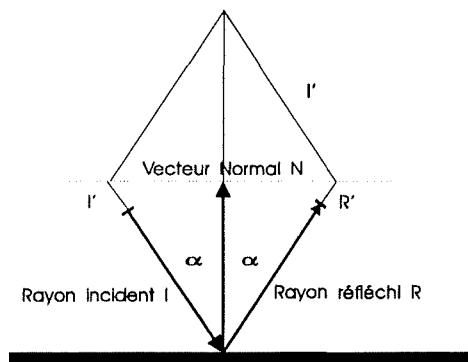


Figure IV.8 : Méthode de calcul du vecteur réfléchi R.

IV. Le rendu réaliste de scènes 3D

En construisant un parallélogramme de diagonale $2*N$, on peut déterminer le vecteur R' dans la direction du rayon réfléchi R à l'aide d'additions vectorielles [FOLE 90]:

$$R' = N + N + I'$$

Le vecteur I' est de même direction que le vecteur incident I . Il n'est cependant pas normé ; sa longueur étant $\frac{1}{\cos \beta}$.

$$I' = \frac{I}{\cos \beta} = \frac{I}{I \cdot N}$$

L'expression complète du vecteur R' , qu'il ne reste plus qu'à normaliser, devient :

$$R' = \frac{I}{I \cdot N} + (2 * N)$$

C. Une autre méthode de calcul du Cosinus β

Le calcul du vecteur R n'est pas indispensable pour évaluer le cosinus de l'angle β [SCHW 87].

En introduisant le vecteur H_i (voir figure IV.9), comme étant la moyenne des vecteurs I et O (c'est-à-dire un vecteur dirigé vers le milieu entre observateur et la source i), on peut calculer le Cosinus de la moitié de l'angle β par :

$$\cos \frac{\beta}{2} = H \cdot N \quad ; \quad \text{avec } H_i = \frac{I + O}{2}, \text{ tous les vecteurs étant normés.}$$

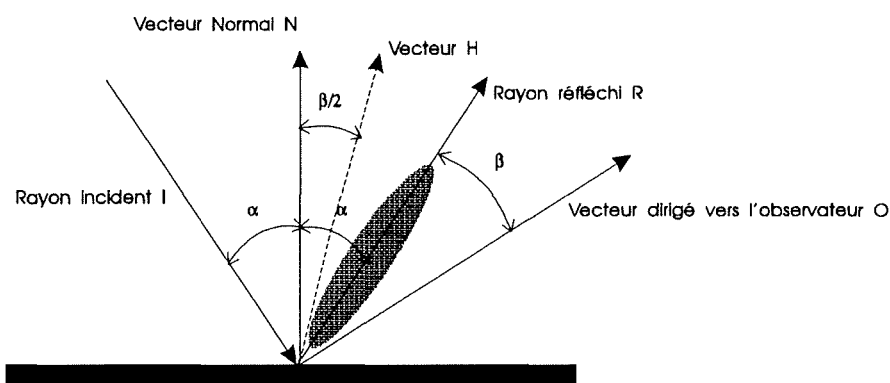


Figure IV.9 : Les vecteurs et les angles relatifs au calcul de la réflexion spéculaire.

Le cosinus de l'angle β s'obtient alors par une simple relation trigonométrique :

$$\cos \beta = [2 * (\cos \frac{\beta}{2})^2] - 1$$

On peut noter que cette dernière opération peut être éliminée en définissant l'angle β comme étant celui entre H_i et N , comme l'a fait à l'origine Blinn [BLIN 77]. Les deux conventions sont acceptables puisqu'il ne s'agit que d'un modèle [SCHW 87].

IV.4.5 Le modèle d'illumination totale

Une unique source de lumière produit un effet peu esthétique, surtout lorsqu'elle est placée au point d'observation, car dans ce cas aucune ombre n'est portée. En pratique, on définit toujours une lumière ambiante pour adoucir les contrastes, et plusieurs sources réparties judicieusement dans la scène. En utilisant plusieurs sources, on peut obtenir une image plus équilibrée, les zones d'ombre étant moins nombreuses (pour autant qu'elles soient calculées) et les contrastes moins forts. On peut noter que dans le cas de sources ponctuelles, une lumière ambiante existe du fait des réflexions diffuses multiples sur les surfaces.

L'intensité lumineuse totale qui arrive à l'observateur depuis un point de la surface résulte donc de plusieurs facteurs, dont un résumé est donné ci-dessous :

L'illumination ambiante :

$$I_{\text{Ambiante}},$$

et l'illumination diffuse et spéculaire provenant des n sources lumineuses :

$$\sum_{i=1}^n I_{\text{Diffuse},i} + I_{\text{Spéculaire},i}$$

La formule générale du **modèle d'illumination totale** en un point de la scène est obtenu par :

$$\begin{aligned} I_{\text{Totale}} &= I_{\text{Ambiante}} + I_{\text{Diffuse}} + I_{\text{Spéculaire}} \\ &= I_a * K_a + \sum_{i=1}^n (I_{\text{Diffuse},i} + I_{\text{Spéculaire},i}) \end{aligned}$$

En considérant les Cosinus par les produits scalaires calculés précédemment, on sait que :

$$I_{\text{Diffuse},i} = K_d * N * L_i * I_i$$

$$I_{\text{Spéculaire},i} = K_s * (N * H_i)^n * I_i$$

Si n sources sont présentes dans la scène, le **modèle d'illumination globale** devient finalement, après simplifications :

$$I_{\text{Totale}} = I_a * K_a + \sum_{i=1}^n (K_d * N * L_i) + \sum_{i=1}^n (K_s * (N * H_i)^n) ,$$

ou encore, si l'on tient compte de l'atténuation en fonction de la distance D de l'observateur aux n sources de la scène :

$$I_{\text{Totale}} = I_a * K_a + \sum_{i=1}^n \frac{I_i}{D_i + M} * [K_d * N * L_i + K_s * (N * H_i)^n]$$

Ce modèle est dû à Phong [PHON 73,75] et a été étendu par Blinn [BLIN 76,77] pour produire des éclats. Whitted [WHIT 80] a, plus tard, ajouté un terme pour les réflexions spéculaires de surfaces parfaitement réfléchissantes (tels les miroirs).

IV.4.6 Le modèle de Cook

Le modèle précédent s'applique assez bien à une grande variété de matériaux. Il suppose cependant que ceux-ci sont parfaits et ne tient pas compte de possibles effets de surface. Cook [COOK 81,82] propose un modèle permettant de calculer l'éclairement de surfaces rugueuses du type le plus général.

L'idée de base est la suivante : relier l'éclat d'un objet à l'intensité et à la taille de chaque source éclairante, en prédisant la distribution directionnelle et la composante spectrale de la lumière réfléchie. Plus précisément, l'intensité de la lumière réfléchie dépend de celle de la source, de la nature et de la capacité de réflexion de la surface éclairée. La composante spectrale de la lumière réfléchie dépend de celle de la source et des caractéristiques de réflexion de la surface.

IV.4.7 Le modèle théorique de Torrance et Sparrow

Le modèle de Phong et Blinn est très réaliste. Mais la composante de réflexion spéculaire n'est pas vraiment exacte.... L'effet est discernable surtout pour des objets non-métalliques. Ce défaut est dû principalement au fait que l'intensité du reflet ne change pas avec la direction de la source, ce qui est incorrect. Ceci est amplifié dans le cas d'une séquence d'animation [THAL 88]. Blinn [BLIN 77] a démontré que le modèle théorique de Torrance et Sparrow [TORR 67] est plus réaliste.

Dans ce modèle, la surface à illuminer est supposée formée d'une collection de très petits miroirs disposés comme des facettes au hasard sur la surface. La composante spéculaire est considérée comme l'ensemble des réflexions venant de toutes ces "micro-facettes" orientées dans la direction du vecteur H_i (ayant la même signification que précédemment). La réflexion spéculaire $I_{\text{Spéculaire}}$ est calculée comme suit :

$$I_{\text{Spéculaire}} = \frac{D * G * F}{N * O}$$

N : vecteur unitaire normal à la surface,

O : vecteur en direction de l'oeil de l'observateur,

D : fonction de distribution des facettes de la surface,

G : facteur qui spécifie les facettes qui s'ombragent et qui se masquent mutuellement,

F : facteur de Fresnel.

Plus de détails sur les paramètres peuvent être trouvés dans Blinn [BLIN 77].

IV.4.8 La répartition de la lumière sur les surfaces

A. Introduction

Le calcul de la répartition de l'illumination sur une surface est complexe et très coûteux en temps calcul. En effet, pour chaque type de modélisation des objets (par facettes, éléments de surfaces Splines,...), l'intensité de la lumière est calculée en tout point des surfaces des objets de la scène avec un des modèles d'illumination vus précédemment. De plus, ces modèles supposent que, en chaque point d'une surface particulière dont on veut déterminer l'illumination, est connu le vecteur normal à cette surface. On voit donc l'intérêt, comme conseillé précédemment, de mémoriser cette information, à priori redondante, dans la structure de données des objets d'une scène.

On perçoit donc bien que ces modèles ne fournissent pas le moyen réel de calculer l'illumination en tout point de la surface, mais seulement l'intensité de lumière en des points spécifiques, que l'on *répartira* ensuite sur l'ensemble de la surface.

Ces techniques de répartition dépendent en fait du type de modélisation des objets.

Si la surface est modélisée par des surfaces Splines, plusieurs méthodes ont été proposées par Cohen [COHE 80] et Catmull [CATM 74].

Si la surface est définie à partir d'un modèle polyédrique, tous les points d'une même facette admettent le même vecteur normal qui est celui du plan de la facette (sauf pour les points situés sur les arêtes). L'intensité des points d'une facette ne sera calculée qu'une seule fois : c'est le **modèle de Lambert** exposé ci-dessous.

B. Le modèle de Lambert

Cette méthode est également connue sous le nom de **modèle de répartition constante**. Ce modèle nécessite un seul calcul d'intensité pour chaque facette. Il implique pourtant les hypothèses suivantes :

- La source de lumière est à l'infini (lumière directionnelle).
- L'observateur est à l'infini.
- Les facettes ne sont pas une approximation d'une surface courbe.

Les deux premières hypothèses sont nécessaires pour assurer que les produits scalaires ($N \cdot L_i$) et ($N \cdot H_i$) soient constants dans les calculs. Comme précédemment, L_i représente le vecteur dirigé vers la source i , N le vecteur normal à la surface, et H_i le vecteur dirigé vers le point milieu entre l'oeil de l'observateur et la source i .

Chaque facette d'un objet va de ce fait avoir nécessairement une intensité lumineuse différente de sa voisine. Ceci est acceptable pour un cube, mais pas pour une sphère d'où la troisième hypothèse. Les vecteurs normaux N , qui sont responsables de cette répartition constante, sont schématisés à la figure IV.10.

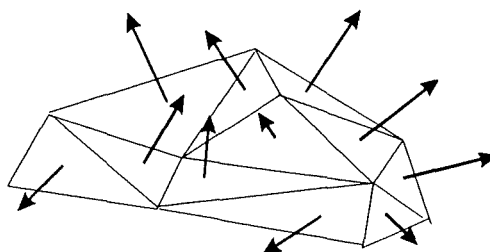


Figure IV.10 : Les vecteurs normaux d'une surface modélisée par facettes planes.

L'éclairage résultant de cette technique, économique en temps calcul, passe par des discontinuités traduisant ces brusques changements de direction de la normale. Ceci est schématisé à la figure IV.11.

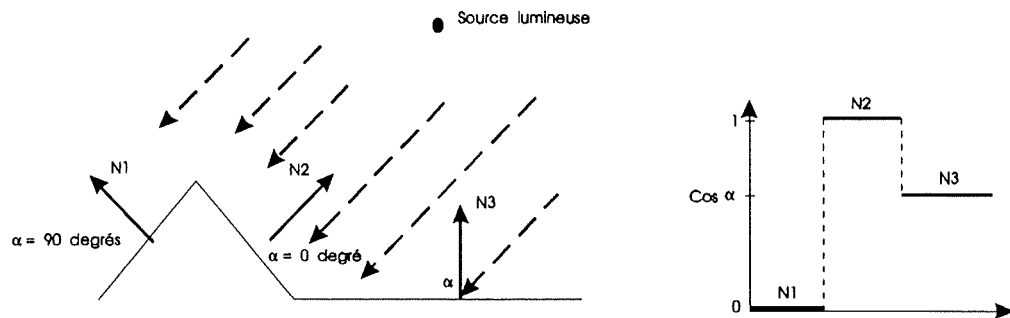


Figure IV.11 : L'éclairage résultant des changements de direction des vecteurs normaux N aux facettes.

Ce phénomène est accentué aux frontières de deux zones d'intensités différentes par un effet de contraste dû à la plus grande sensibilité de l'oeil aux différences relatives plutôt qu'aux différences absolues.

Ce premier modèle permet déjà d'obtenir une bonne approximation de l'illumination réelle d'une scène modélisée par des facettes planes. L'illumination de chaque facette est constante et est souvent appelée l'illumination "*flat shading*". Cette méthode considère que toutes les faces sont mates, sans reflets.

Pour la modélisation polyédrique, différentes *techniques de lissage* permettent d'atténuer ces discontinuités. Trois méthodes de lissage très répandues sont utilisées : le **lissage de Gouraud** et son extension, le **lissage de Phong**.

C. Le lissage de Gouraud

Henri Gouraud [GOUR 71] propose d'éliminer les discontinuités du modèle de Lambert par une *interpolation bilinéaire*.

Cette méthode consiste à calculer les vecteurs normaux aux sommets des facettes, puis de déterminer l'illumination en ces points. Celle-ci rend alors les facettes imperceptibles.

Le vecteur normal en un sommet est évalué en prenant la moyenne des vecteurs normaux des facettes se joignant en ce point, comme schématisé dans la figure IV.12.

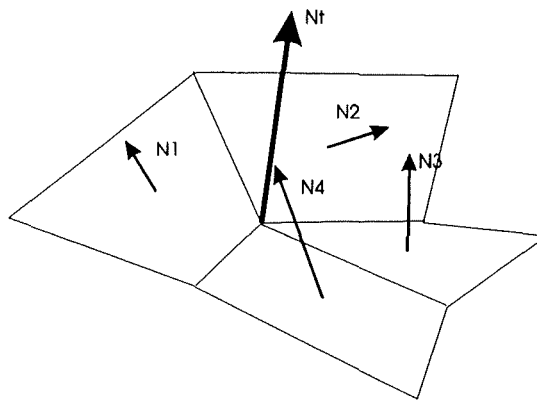


Figure IV.12 : Evaluation d'un vecteur normal N_t en un sommet.

Pour colorier ensuite l'intérieur de la facette, on utilise une interpolation bilinéaire, appelée **algorithme Scan-Line**, et décrite ci-dessous.

Soit L une ligne de balayage (voir figure IV.13) coupant deux arêtes M_1M_2 et M_3M_4 en deux points A et B définis par :

$$A = M_1 + \alpha (M_2 - M_1) \text{ et } B = M_3 + \beta (M_4 - M_3)$$

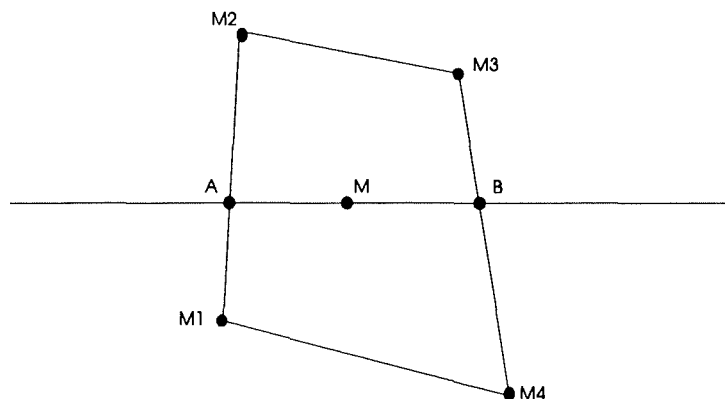


Figure IV.13 : Ligne de balayage de l'algorithme Scan-Line.

Les couleurs (symbolisées par K) en A et B sont calculées par :

$$K_A = K_1 + \alpha (K_2 - K_1)$$

$$K_B = K_3 + \beta (K_4 - K_3)$$

Le coloriage des points $M = A + \theta (B - A)$ du segment AB est donné par l'équation :

$$K_M = K_A + \theta (K_B - K_A)$$

Cette technique de lissage donne des résultats satisfaisants, mais n'est cependant pas parfaite. En effet, lorsque deux normales successives sont parallèles, il faut essayer de subdiviser la facette pour éviter que de grandes zones soient considérées comme étant de luminosité uniforme alors que la surface présente des variations.

En calculant la normale en un sommet à partir d'une seule face, on peut former une discontinuité de luminosité pour représenter une arête vive.

Avant de continuer, résumons succinctement le principe du lissage de Gouraud ; ce dernier étant à la base de la seconde technique de lissage exposée ci-après.

1) Pour chaque facette i de la surface, une normale N_i est calculée comme un vecteur perpendiculaire au plan de celle-ci.

2) Pour chaque sommet commun à n facettes, une unique normale N_t est calculée comme moyenne des normales N_i de ces n facettes ;

$$N_t = \frac{\sum_{i=1}^n N_i}{n}$$

3) Les intensités à ces sommets sont calculées en utilisant les normales N_t aux sommets et un des modèles vus précédemment.

4) Comme chaque facette fournit une illumination différente à chaque sommet, l'illumination en un point intérieur d'une facette est trouvée par une interpolation linéaire des intensités des sommets le long de chaque arête et entre les arêtes le long de chaque ligne de balayage.

D. Le lissage de Phong

Dans la méthode de Gouraud, l'illumination était calculée aux sommets, puis interpolée à travers la facette. Phong propose lui d'interpoler le vecteur normal aux facettes au lieu de l'interpolation des intensités. On calcule ensuite l'intensité des points en appliquant un modèle d'illumination pour chaque point. On obtient une bien meilleure approximation de la courbure de la surface et les reflets (la réflexion spéculaire) sont mieux rendus.

Cette technique est cependant très gourmande en temps calcul ; on doit en effet recalculer le vecteur normal, le normaliser, et puis seulement appliquer un modèle d'illumination. Avec les mêmes notations que précédemment, on applique les calculs d'interpolation suivants (voir figure IV.14):

$$N_a = N_1 + \alpha (N_2 - N_1)$$

$$N_b = N_3 + \beta (N_4 - N_3)$$

$$N_m = N_1 + \theta (N_b - N_a)$$

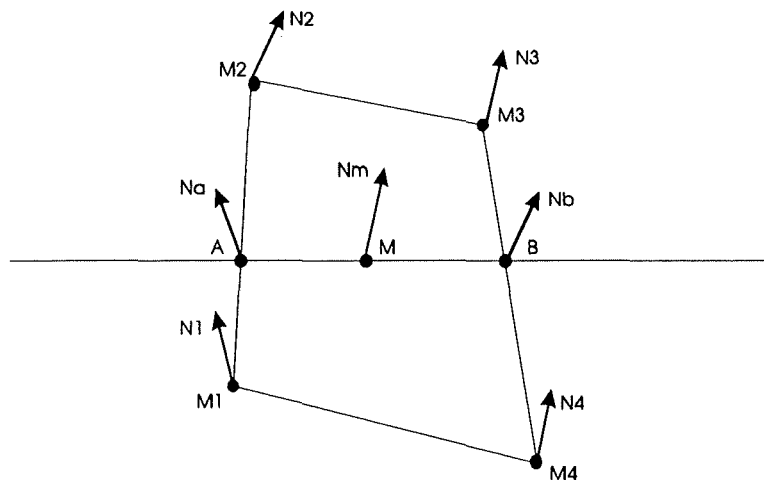


Figure IV.14 : Ligne de balayage pour l'algorithme Scan-Line.

Des sophistications peuvent être apportées à cette méthode pour assurer la continuité de l'illumination et pour éviter les variations rapides désordonnées dans une séquence d'animation [FOLE 90].

E. Comparaison des méthodes

Dans le modèle de répartition constante de Lambert, une intensité uniforme est attribuée à chaque face. On obtient donc des changements d'intensité brutaux aux jointures des facettes. Cette manière de faire n'est pas acceptable, car l'oeil va discerner très nettement ces frontières.

Tout l'intérêt des méthodes d'interpolation de Gouraud et de Phong est qu'elles peuvent être utilisées conjointement avec les algorithmes de remplissage des figures par balayage (qui reposent sur les mêmes principes *Scan-Line*) [NEWM 79].

IV. Le rendu réaliste de scènes 3D

Il est important de remarquer que les calculs impliqués par le lissage de Phong sont plus lourds que ceux de Gouraud. En effet, pour chaque point à afficher, il est nécessaire de déterminer le vecteur normal unitaire, c'est-à-dire de le diviser par sa longueur.

$$\mathbf{N} = \frac{\mathbf{N}}{||\mathbf{N}||} ; \text{ avec } ||\mathbf{N}|| = \sqrt{N_x^2 + N_y^2 + N_z^2}$$

Ces calculs prennent un temps calcul considérable ; cette expression implique en effet trois multiplications, deux additions, une extraction d'une racine carrée, et une division, et ceci, soulignons-le encore, pour chaque point de la surface à afficher !

Le lissage de Gouraud ne fait appel qu'à des interpolations bilinéaires ; il est donc plus simple et plus rapide. Il manque cependant des variations d'illumination quand le vecteur normal à la surface traverse la direction de la lumière. En conséquence, l'effet spéculaire n'a lieu que pour des positions bien précises de l'oeil.

Les schémas de la figure IV.15 montrent l'évolution de l'éclairement d'une surface selon les deux techniques de lissage.

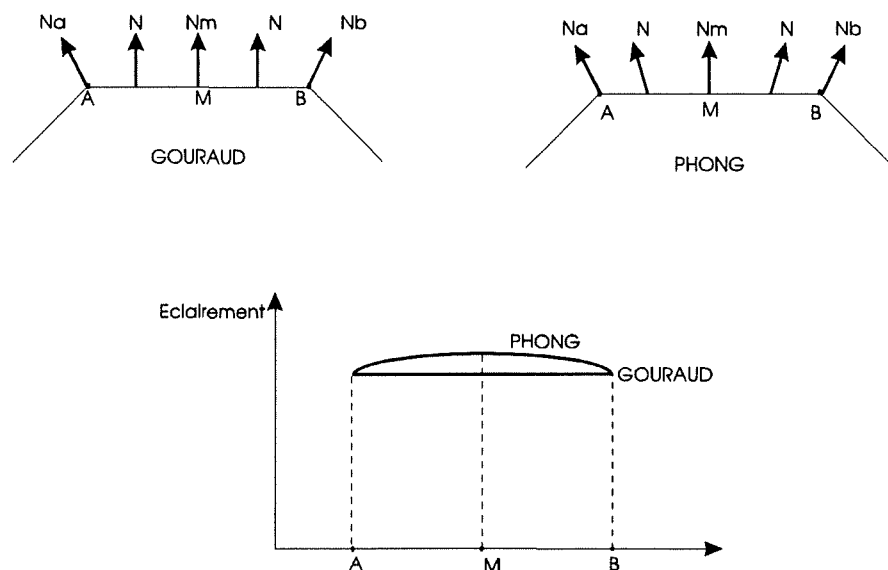


Figure IV.15 : Evolution de l'éclairement en fonction des techniques de lissage.

Les deux méthodes présentent un même défaut (représenté à la figure IV.16) quand les facettes sont concaves ou contiennent des trous. Il est possible, par triangularisation, d'éviter de telles configurations [BRET 88]. C'est notamment pour cette raison que, lorsque l'on entame une modélisation polyédrique d'une scène, on préférera généralement l'utilisation de facettes triangulaires.

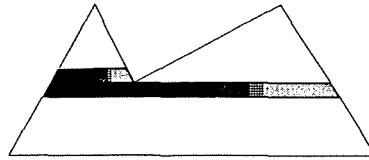


Figure IV.16 : Défaut de répartition de l'intensité lumineuse lors de la présence de facettes concaves.

IV.4.9 Conclusion

Les modèles d'illumination que je viens d'exposer permettent de déterminer l'aspect de l'éclairement en chaque point d'une scène. Les résultats de leur usage ne produisent cependant pas une scène *réaliste*. Il faut adjoindre des *effets spéciaux* à ces modèles qui augmenteront le réalisme de la scène affichée. Je propose dès lors de développer ci-après le rendu de la **transparence**, des **ombres**, et des **textures** des surfaces.

Il existe cependant deux algorithmes qui sont capables de produire, au prix de temps calcul énormes, la plupart des effets d'illumination d'une scène avec un réalisme étonnant le **lancer de rayons**, et la méthode dite de **radiosité**, étudiés dans la suite de ce chapitre.

IV.5 La transparence

Généralement, quand un objet est devant un autre, il le masque. On peut simuler cela par l'application d'un algorithme de détermination des parties cachées, en supposant que tous les objets sont opaques. Il existe cependant des objets naturellement transparents (comme le verre, le plastique,...) qui laissent passer la lumière. Ces objets transparents doivent pouvoir être reproduits de manière synthétique si l'on désire un rendu réaliste convaincant.

Les objets situés derrière les objets transparents ne doivent plus être cachés. Il faut alors représenter différemment ces objets vus à travers des objets transparents.

Plusieurs techniques de rendu de la transparence ont été mises au point. Les plus évoluées se basent sur les lois physiques de la réfraction, comme la **loi de Snell-Descartes**. Tenir compte de ces lois implique une consommation considérable en temps calcul. Si on peut se permettre moins de réalisme, il existe des techniques moins complexes, comme la **méthode du peintre**.

IV.5.1 La méthode du peintre

La transparence peut être implémentée simplement en affichant les surfaces selon la méthode du peintre : chaque nouvelle surface est peinte à l'aquarelle, de manière à laisser passer par transparence les surfaces précédentes. Prenons l'exemple d'une boîte rouge vue à travers une vitre verte. La couleur de la boîte doit être un mélange des deux. La proportion de rouge et de vert dépendra du coefficient de transmission de la vitre. Si elle est complètement opaque, la boîte sera colorée en vert, si elle est parfaitement transparente, elle sera rouge. Si elle est entre les deux, on obtiendra un mélange de rouge et de vert.

L'éclairement en un point d'une surface transparente est alors calculé comme combinaison linéaire de la lumière qu'il réfléchit et de celle réfléchi par un objet immédiatement situé derrière lui. Par conséquent, avant d'afficher un nouveau point de la scène, on multipliera simplement l'intensité du point actuel par un facteur correspondant à la transparence K_t (compris entre 0 et 1) de la surface à afficher puis on lui additionnera la luminosité du point à afficher diminuée en proportion inverse :

$$I = I_1 * K_t + I_2 * (1 - K_t)$$

I_1 correspond à l'intensité lumineuse du point 1 situé derrière le point 2, I_2 à l'intensité lumineuse du point 2. K_t symbolise le facteur de transparence de la facette contenant le point 2, c'est-à-dire la facette à afficher.

La valeur du facteur K_t varie de 0 (facette complètement transparente) à 1 (complètement opaque).

Cette formule peut être appliquée de manière récursive lorsqu'une succession d'objets sont vus par transparence.

IV.5.2 La loi de Snell-Descartes

Une technique évoluée de simulation de la transparence tiendrait compte exactement des lois physiques et plus particulièrement de la loi de Snell-Descartes, qui spécifie que le rayon lumineux traversant la surface de séparation de deux milieux d'indices de réfraction η_1 et η_2 subit une déviation obéissant à un phénomène de réfraction, résumé selon les deux points suivants :

1) Le rayon incident I , le vecteur normal N au point d'incidence et le rayon réfracté (ou transmis) T sont dans un même plan.

2) L'angle du rayon réfracté δ (voir figure IV.17) dépend de l'angle du rayon incident α et des coefficients de réfraction des deux milieux, η_1 et η_2 .

$$\eta_1 \cdot \sin \alpha = \eta_2 \cdot \sin \delta$$

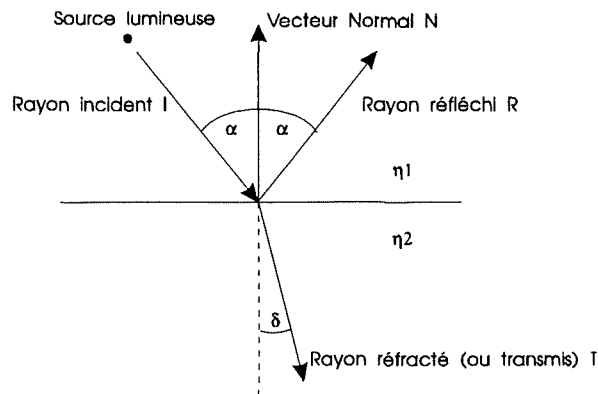


Figure IV.17 : La réfraction de la lumière lors d'un changement de milieu.

En plus du rayon transmis T, il faut aussi, comme dans le cas d'un matériau opaque, considérer un rayon réfléchi R car toute l'énergie incidente n'est pas transmise ou absorbée (voir de nouveau la figure IV.17). La proportion de l'intensité réfléchie et transmise dépendra des coefficients de transmission et de réflexion de la surface. La somme de ces deux coefficients sera toujours plus petite que 1.

On peut noter enfin que, si on veut réellement tenir compte de tous les phénomènes physiques tels que la réfraction, l'algorithme du lancer de rayons reste, de loin, le plus efficace.

IV.6 La détermination des ombres

Les sources de lumières jouent un rôle fondamental dans le réalisme. Mais il faut remarquer que, à moins de placer une unique source de lumière à l'emplacement de l'observateur, il est indispensable de tenir compte des ombres. La présence d'ombres dans une scène permet de mieux percevoir encore les volumes dans l'espace. Elles informent sur les positions relatives des objets et des lumières.

IV.6.1 Les types d'ombres

Pour calculer correctement les ombres présentes dans l'illumination d'une scène, il importe de considérer les deux types d'ombres : les *ombres propres* et les *ombres portées*.

Un objet se divise en une partie éclairée et une partie à l'ombre, appelée l'ombre propre, c'est-à-dire la partie cachée par rapport à la source. Les ombres propres se déterminent donc par des algorithmes de recherche des facettes arrière [FOLE 90].

Dire qu'un point est dans une zone d'ombre portée, c'est dire qu'il ne voit pas la source, et que celle-ci, considérée comme un oeil ne voit pas ce point. Ainsi, la résolution du problème des ombres portées s'apparente à celui de l'élimination des parties cachées.

Ces deux types d'ombre sont schématisés à la figure IV.18.

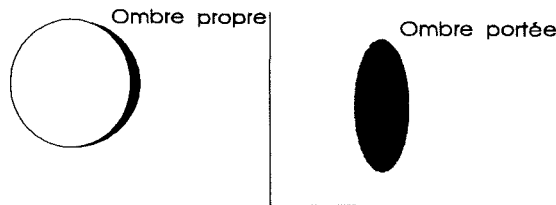


Figure IV.18 : Les deux types d'ombres.

Le traitement des ombres peut être très complexe s'il n'est pas simplifié. De plus, lors de la présence de sources *étendues* (non ponctuelles), des zones de pénombre sont à considérer, comme dans la figure IV.19, alors que des sources ponctuelles génèrent des ombres nettes.

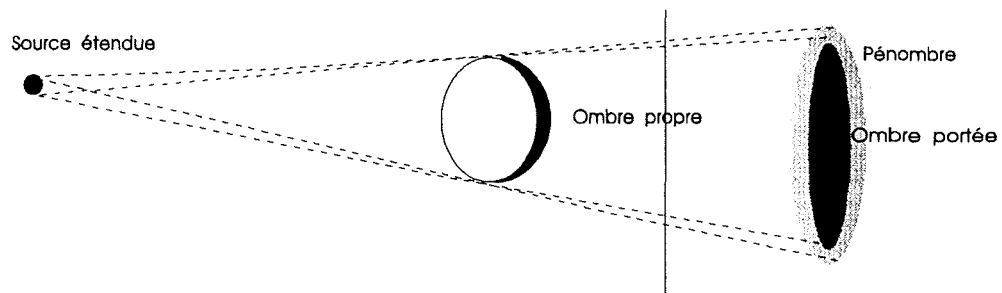


Figure IV.19 : Zone de pénombre due à une source non ponctuelle.

Le rendu des zones de pénombres dues à des sources étendues peut s'obtenir en appliquant le modèle d'illumination de Cook [COOK 81,82]. Mais en pratique, on se limite généralement à des sources ponctuelles placées à l'infini pour déterminer les ombres portées.

IV.6.2 Les techniques de production d'ombres portées

On peut classer les techniques de production d'ombres portées en quatre catégories.

1) Les techniques basées sur un *algorithme de découpage de polygones* ; les deux principaux algorithmes sont ceux de Nishita et Nakamae [NISH 74], et de Atherton [ATHE 78].

2) La technique des **volumes d'ombres** introduite par Crow [CROW 77].

3) Les techniques basées sur un algorithme de mémoire de profondeur ("**Z-Buffer**"). Les deux principaux algorithmes sont ceux de Williams [WILL 78], et de Brotman et Badler [BROT 84].

4) La technique du **lancer de rayons**.

Examinons brièvement le principe des trois dernières techniques. Celles-ci étant les plus utilisées en pratique.

A. La technique des volumes d'ombre

Par rapport à une source lumineuse, une facette d'un objet définit une région de l'espace qu'elle obscurcit, appelée *volume d'ombre*, et qui est théoriquement illimitée. Mais, si l'on considère l'intersection de ce volume d'ombre avec le volume de vision de l'observateur, on obtient un volume convexe (voir à la figure IV.20). Ce dernier sera alors traité comme étant imparfaitement transparent lors du processus d'affichage.

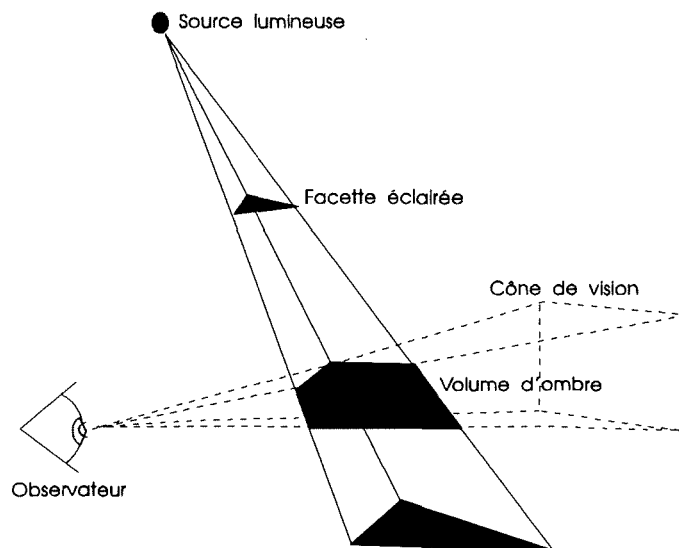


Figure IV.20 : La technique des volumes d'ombres.

L'algorithme original de traitement de ce volume d'ombre est dû à Crow [CROW 77], et peut être intégré dans divers algorithmes d'élimination des parties cachées. Il comporte cependant des restrictions concernant les objets représentés. Fortin et al. ont décrit des moyens de supprimer ces limitations.

Quelques remarques concernant cette technique sont à souligner :

- L'obscurcissement du volume destiné à représenter l'ombre dépend de l'éloignement de la source lumineuse et de son intensité.

- La zone ombragée reçoit de la part de l'environnement de la lumière ambiante, et n'est donc pas complètement noire.

- Dans le cas de sources multiples, différentes zones d'ombres plus ou moins intenses apparaissent car un point dans l'ombre d'une source peut recevoir de la lumière d'autres sources. En cas d'ombres multiples, on veillera à additionner les obscurcissements.

B. La technique du Z-Buffer

Lors de la détermination des parties cachées étudiée précédemment, l'algorithme du Z-buffer peut être adapté pour calculer les ombres portées.

Il suffit d'utiliser un second buffer pour stocker les profondeurs des points éclairés par une source (c'est-à-dire vus depuis celle-ci). Un algorithme en pseudo-code peut se présenter comme suit :

Pour chaque pixel (x,y) visible depuis l'oeil,

Calculer ses coordonnées (x',y',z') dans le repère de la source,

La valeur z' est ensuite comparée à la valeur z'' stockée en (x',y') du Z-Buffer de l'ombre.

Si $z' < z''$, le point est éclairé.

Sinon, il est dans l'ombre.

Un inconvénient majeur est évident : il faut réserver autant de "Z-Buffers" qu'il y a de sources lumineuses. D'où un problème de réservation de mémoire important !

C. La technique du lancer de rayons

La méthode du lancer de rayons exposée lors du calcul des parties cachées peut également être adaptée pour déterminer les ombres portées dans une scène.

Le principe est le suivant : un rayon issu de l'oeil balaye l'écran et, en chaque pixel de celui-ci, sont calculées les intersections de ce rayon avec la scène. A partir du point visible (c'est-à-dire celui qui est le plus proche de l'oeil) sont lancés autant de rayons qu'il y a de sources lumineuses en direction de celles-ci. Ce point sera dans l'ombre d'une source, si le rayon correspondant intersecte un élément de la scène. L'éclairement du point est alors obtenu en additionnant les influences des sources qui l'illuminent.

La détermination des ombres est très coûteuse en temps calcul. Aussi, utilise-t-on souvent une lumière très diffuse (tel le ciel) pour limiter le manque d'ombres. On peut remarquer également que tous les algorithmes de détermination d'ombres sont limités, sauf le célèbre et très coûteux lancer de rayons, dont je reparlerai plus en détail dans la suite de ce chapitre.

IV.7 Les Textures

IV.7.1 Les techniques de textures

Les objets synthétisés par ordinateur ont souvent cet aspect lisse et plastique, qui leur donne une apparence artificielle. Ce problème devient très gênant par exemple dans le cas de synthèse d'êtres humains, dont la peau paraîtra trop parfaite et donnera l'impression de mannequin de cire [THAL 88]. On peut réduire cet aspect artificiel en appliquant sur la surface certains détails.

Il y a deux catégories de détails à appliquer sur une surface : ceux correspondant à la répétition d'un motif (on réalise ce que l'on appelle couramment un "*Pattern Mapping*"), et ceux destinés à simuler une certaine rugosité (comme la peau d'une orange par exemple). Dans les deux cas, on parle de simulation de **textures** [BLIN 76], définies comme étant une *microstructure* du détail à représenter.

Différentes techniques de synthèse de textures sont possibles. Elles peuvent être d'apparence structurée et faire alors appel à des modèles déterministes, ou d'apparence hasardeuses et font alors référence à des modèles stochastiques :

Pour les modèles déterministes, il existe les techniques suivantes :

- L'application d'une image à deux dimensions sur la surface de l'objet - souvent appelé un "*Texture Mapping*".
- La perturbation de la lumière sur la surface à l'aide d'une *fonction de texture*.

Pour les modèles stochastiques, on retient principalement deux techniques :

- La simulation de relief par la *technique des fractals*.
- La simulation de corps "flous" par les *systèmes de particules*.

Les objets fractals ont été imaginés par Benoît Mandelbrot [MAND 75] pour rendre compte des propriétés de certains objets naturels extrêmement complexes et irréguliers tels les montagnes, l'écorce d'un arbre, les cheveux, ...

Les systèmes de particules de Reeves [REEV 83] permettent la modélisation de phénomènes naturels, et d'objets "flous" de forme irrégulière, mal définie et qui évolue au cours du temps tels les nuages, le feu, la fumée, l'eau, l'explosion, les tissus, ...

Ces techniques font l'objet de nombreuses recherches et une importante littérature à été publiée à ce sujet. Elles sont complexes, et il est malheureusement (tant elles sont ingénieuses !) impossible de les aborder dans le cadre de ce mémoire, limité en nombre de pages. Je me limiterai donc à décrire brièvement les deux méthodes de textures déterministes très fréquemment employées.

IV.7.2 L'application d'une image 2D

Ce procédé consiste à se définir en mémoire un modèle standard de la microstructure en question et à la *coller* (on utilise également le terme *mapper*) sur la surface.

En découpant simultanément la surface et la texture, on peut réaliser des correspondances région par région plutôt que point par point. Cette technique de subdivision récursive est due à Catmull [CATM 74]. Cela revient à découper la texture selon une grille puis à faire une correspondance entre chaque case de la grille et son équivalent sur la surface. On transforme alors la partie de la texture contenue dans la case selon la forme de la facette équivalente (voir figure IV.21).

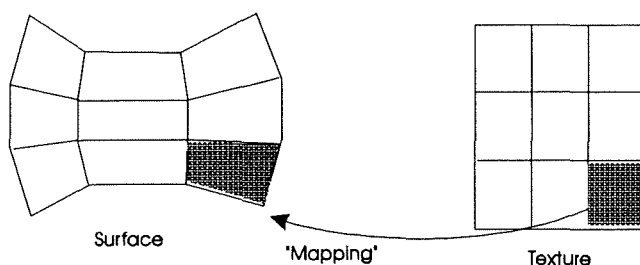


Figure IV.21 : Mapping d'une texture sur une surface.

Une application de cette technique de "mappage de textures" consiste à numériser une surface de texture réelle (par exemple, l'écorce d'un arbre), et de la *mapper* ensuite sur la surface synthétique (par exemple, un cylindre). Mais cette manière de procéder n'est pas tout à fait satisfaisante. En effet, la texture numérisée n'est qu'une projection 2D d'une structure 3D, d'où l'impression de peinture, d'un collage photographique obtenu.

IV.7.3 La perturbation de la lumière

Nous avons vu précédemment que le calcul de l'intensité lumineuse en un point dépend du vecteur normal en ce point. Blinn [BLIN 76] propose de modifier judicieusement ce vecteur afin de créer des zones de lumière et d'ombre qui simulent une texture. Pour perturber ce vecteur normal, on utilise une fonction mathématique à deux arguments $F(x,y)$ qui est censé mesurer le déplacement de la surface irrégulière par rapport à la surface idéale (voir figure IV.22). Un nouveau point P' sur la surface perturbée est obtenu par l'équation :

$$P' = P + (F * N) \quad ; \quad P \text{ étant le point original et } N \text{ le vecteur normal.}$$

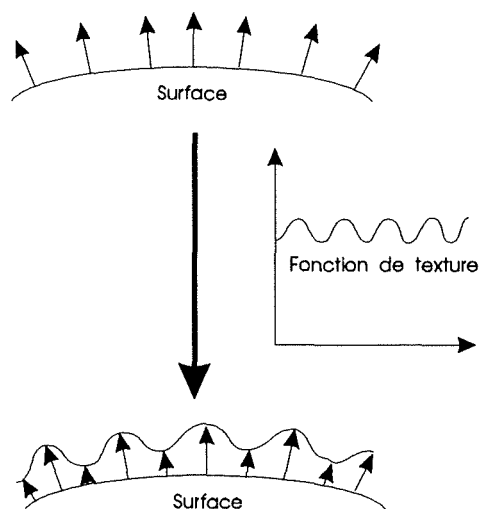


Figure IV.22 : La perturbation des vecteurs normaux par une fonction de texture.

Cette technique a été généralisée avec des fonctions tridimensionnelles $F(x,y,z)$. Elle porte alors le nom de "*texture solide*" (Perlin [PERL 85], Peachey [PEAC 85], Brossard et al. [BROS 85]). A chaque point de la surface correspond un point dans l'espace de texture. Le résultat de ce procédé est d'un réalisme étonnant, comme le démontre l'illustration I à la fin de ce chapitre.

Dans la pratique, le choix d'une fonction de texture est délicat, et souvent déterminé empiriquement. Blinn a proposé quelques exemples de fonctions intéressantes dans [BLIN 76].

IV.8 Le lancer de rayons

IV.8.1 Introduction

La méthode du lancer de rayons - le "ray tracing" - est le moyen le plus simple et à la fois le plus complet pour générer des images de synthèse réalistes. Elle est basée sur la simulation numérique des lois de l'optique géométrique. Elle permet de réaliser l'élimination des parties cachées, d'implémenter aisément la réflexion, la transparence et la réfraction des rayons lumineux, ainsi que de traiter les ombres et les pénombres.

Le principal avantage du lancer de rayons sur d'autres techniques est de pouvoir reproduire les réflexions d'objets sur d'autres, et ce même pour des réflexions multiples, au lieu d'uniquement rendre les reflets des sources lumineuses sur les objets de la scène.

Le terme de *lancer de rayon*, souvent utilisé, n'est pas parfaitement approprié. On pourrait lui préférer le terme de *suivi de rayon* qui fait mieux ressortir, comme nous le verrons, le principe de cet algorithme.

IV.8.2 Le parcours des rayons lumineux

Intuitivement, on peut considérer que l'image d'une scène perçue par l'observateur est formée de l'ensemble des rayons lumineux qui atteignent son oeil. Ces rayons proviennent des sources de lumière directement ou indirectement, après réflexion sur les objets de la scène.

La figure IV.23 illustre les cheminements complexes que parcourent les rayons lumineux pour atteindre l'observateur.

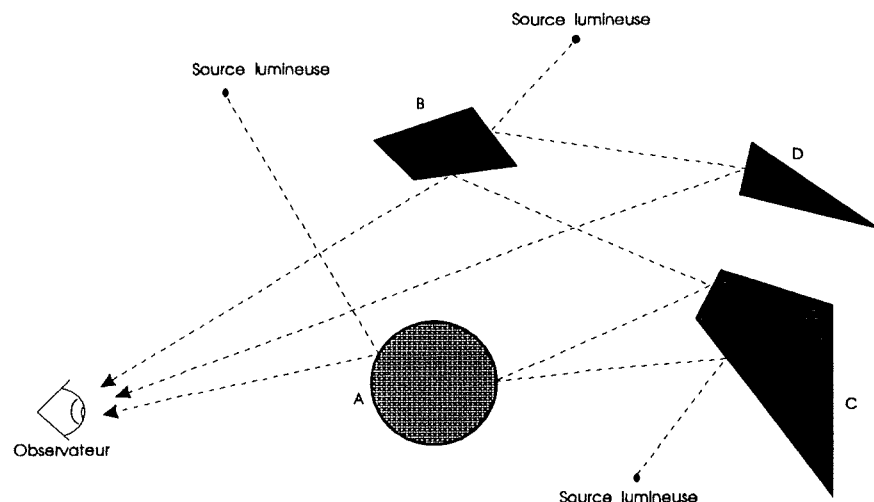


Figure IV.23 : Le parcours des rayons lumineux.

Un objet (dans la figure IV.23, l'objet C) peut par exemple être masqué pour l'observateur par un autre objet (A), mais peut cependant être vu par réflexion sur un autre objet (B) de la scène. De même, la face arrière d'un objet (B) peut être visible par réflexion sur d'autres (D). Cette dernière considération signifie que les parties cachées ne doivent pas être éliminées avant d'utiliser la méthode du lancer de rayons. En fait, comme je l'ai déjà signalé, l'élimination des parties cachées est réalisée automatiquement par l'algorithme du lancer de rayons.

IV.8.3 Le Principe

L'image de la scène que l'observateur perçoit peut être considérée comme un écran placé entre la scène et lui-même. On peut alors s'imaginer que pour chaque pixel de l'écran, un seul rayon atteint l'oeil. On pourrait dès lors considérer tous les rayons émis par les sources pour voir lesquelles atteignent l'oeil, et calculer l'intensité lumineuse pour tous les points de l'image.

Cette approche serait catastrophique voire même impossible car seuls quelques rayons arrivent à l'observateur. Un artifice est donc nécessaire...

D'après le principe du retour inverse de la lumière, le chemin d'un rayon lumineux issu d'une source ponctuelle S et éclairant un point M est le même que si la source était en M et le point éclairé en S. Ainsi, au lieu de chercher le chemin des rayons lumineux *arrivant* à l'oeil, on procède dans l'autre sens. La méthode du lancer de rayons consiste alors à *lancer*, depuis l'oeil, autant de rayons qu'il y a de points sur l'écran à calculer et, pour chacun d'eux, on remonte l'histoire de leurs multiples réflexions et réfractions à travers la scène.

Lorsque le rayon atteint un objet et que c'est l'objet le plus proche de l'oeil, on appliquera les lois de la réflexion et, si l'objet est transparent, les lois de la réfraction, afin de calculer de quelles directions proviennent les deux rayons qui vont former celui qui a atteint l'oeil. On répète, de manière récursive, ce *suivi* de parcours inverse de la lumière jusqu'à arriver à l'infini ou à une source lumineuse (dont fut issu le rayon). A ce point, la luminosité est connue, ce qui permet lors du retour de la récursion de calculer la luminosité du rayon au dernier changement de surface. Pour ce faire, il suffit de sommer les intensités du rayon réfléchi et du rayon transmis. Par ce processus récursif, on peut évaluer l'intensité du rayon reçu par l'oeil de l'observateur. Cette intensité correspond à la luminosité du pixel de l'écran associé. L'intensité en chacun de ces pixels sera atténuée d'un facteur dépendant de la longueur du rayon correspondant.

La méthode du suivi de rayon revient donc à construire un arbre binaire regroupant tous les rayons qui sont à l'origine de celui qui atteint finalement l'oeil. Le calcul d'illumination correspond au parcours dans l'ordre inverse de cet arbre en évaluant le modèle d'illumination à chaque changement de surface. Par une implémentation récursive, on évite une gestion explicite de l'arbre, ainsi que son stockage en mémoire, vu que seule la branche que l'on parcourt est intéressante.

La racine de l'arbre binaire (schématisé à la figure IV.24) représente l'oeil de l'observateur et les noeuds de niveau 1, l'intersection la plus proche de l'oeil des rayons joignant celui-ci à chaque pixel. Deux branches partent de chaque noeud de niveau 1 ; un pour le rayon de réflexion spéculaire R et un pour le rayon réfracté T, s'il existe. Les noeuds de niveau 2 représentent les intersections les plus proches des noeuds de niveau 1 avec les objets de la scène, etc ...

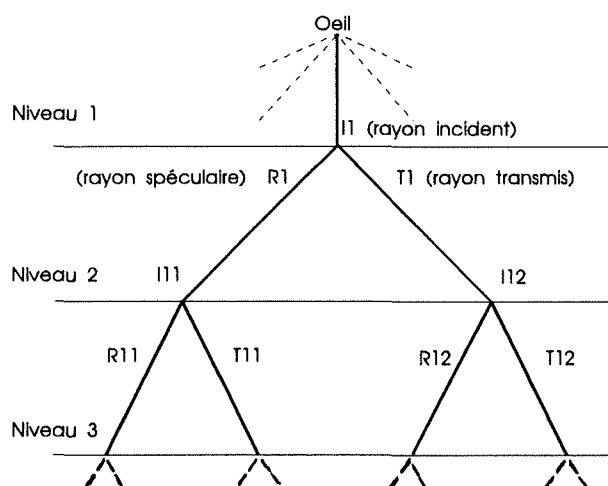


Figure IV.24 : L'arbre binaire du processus récursif du lancer de rayons.

IV.8.4 L'Algorithme du Lancer de Rayons

KAY [KAY 79] et Whitted [WHIT 80] ont été les premiers à implémenter un algorithme de lancer de rayons, initialement suggéré par Appel [APPE 68].

On peut le résumer, à un niveau d'abstraction élevé, de la façon suivante :

Pour chaque pixel de l'écran

Faire

Calculer la direction du rayon partant de l'observateur passant par ce pixel

Appeler la procédure récursive de calcul de la luminosité de ce rayon

Fait

IV. Le rendu réaliste de scènes 3D

La procédure récursive du calcul de la luminosité du rayon comportera les paramètres suivants : la *profondeur de la récurrence*, la *direction du rayon suivi*, la *transformation courante*, et la *luminosité calculée* (paramètre résultat).

La profondeur de récurrence est souvent limitée afin de diminuer le temps de calcul, au détriment de la précision. Pour une profondeur maximale de 1, l'algorithme réalise en fait l'élimination des parties cachées et l'illumination selon le modèle de Phong. La transformation courante est utilisée pour diriger l'axe Z parallèlement à la direction du rayon suivi, de manière à simplifier les calculs d'intersection [FOLE 90]. Le dernier paramètre sert à retourner le résultat, c'est-à-dire l'intensité lumineuse calculée.

La procédure récursive s'arrête lorsqu'un rayon n'intersecte aucun objet. Mais ce cas peut ne jamais se produire comme en témoigne l'exemple de deux miroirs parallèles. Un *test d'arrêt* est donc nécessaire. On peut choisir plusieurs types de tests :

- Soit une profondeur de récurrence arbitrairement fixée a priori.
- Soit un seuil minimal de l'intensité lumineuse calculée. Celle-ci s'atténuant en fonction du niveau, on peut décider qu'à partir d'une certaine valeur, sa contribution devient négligeable.
- Soit le processus est poussé jusqu'à la limite permise par la taille de la pile disponible en mémoire.

La procédure récursive peut être décrite comme suit :

Faire

Si la profondeur maximale de récurrence est atteinte, retourner la luminosité ambiante, sinon l'incrémenter

Rechercher toutes les intersections du rayon avec les objets
Si aucune intersection n'existe, retourner la lumière ambiante

Pour l'objet dont l'intersection est la plus proche de l'observateur

Faire

Calculer la direction du rayon réfléchi

Appeler récursivement la procédure sur ce rayon

Si l'objet est transparent

Faire

Calculer la direction du rayon transmis

Appeler récursivement la procédure sur ce rayon

Fait

Fait

Calculer la luminosité totale.

Fait

IV.8.5 Les tests d'intersection

L'examen de l'arborescence associée à la méthode du lancer de rayons montre que le nombre de calculs augmente très rapidement d'abord avec la résolution de l'image (nombre de pixels à considérer), ensuite avec le nombre de primitives traitées (nombre d'objets dans la scène) et enfin avec la profondeur de récurrence.

La détermination des intersections des rayons avec les objets de la scène est le point le plus délicat de l'algorithme du lancer de rayons. La plus grande partie de l'effort de calcul est investie dans ces tests d'intersection. En effet, tous les objets de la scène doivent être pris en considération pour chaque rayon. Ces calculs d'intersection sont très fréquents et très gourmands en temps calcul. Il convient dès lors de les optimiser judicieusement, comme par exemple, en considérant les sphères englobantes des objets, vu que le calcul d'intersection d'un rayon et d'une sphère est beaucoup plus simple. Il suffit dans ce cas de tester la distance du rayon tracé par rapport au centre de la sphère. Si cette distance est plus petite que le rayon de la sphère, il est évident qu'il y a intersection, qu'on obtient facilement en combinant les équations de la droite représentant le rayon et les équations de la sphère.

C'est notamment pour cette raison que les images les plus caractéristiques produites par la technique du lancer de rayons comportent des volumes construits par assemblage de sphères polies, laissant apparaître en réflexion d'autres objets. (voir photo X en annexe)

Pour d'autres formes géométriques, le test est plus complexe. Dans le cas d'une face plane, on calculera l'intersection avec le plan la supportant. On vérifiera ensuite que cette face est bien interne à la projection sur le plan de l'image. Cette technique est assez générale ; on travaillera plutôt sur une projection que sur le volume lui-même.

On évitera d'utiliser des méthodes compliquées pour le calcul des intersections. Il est de loin préférable d'utiliser des tests de discrimination simples et rapides. Par exemple, pour chaque objet, on peut définir sa sphère englobante et commencer par tester si le rayon l'intersecte. Cette méthode peut être étendue à un groupe d'objets rapprochés, ou pour tester l'intersection avec chacune des parties d'un objet. En définissant toute une hiérarchie de sphères englobantes, on peut éliminer très rapidement des cas triviaux de non-intersection.

L'utilisation de tests plus sophistiqués est rare. Le calcul du point d'intersection n'est simple que pour la sphère, la facette plane, le cylindre, et les volumes de révolution.

On peut noter que Kajiya [KAJI 84] a mis au point de nouvelles méthodes pour déterminer les intersections des rayons avec divers types d'objets particuliers.

IV.8.6 La direction des rayons réfléchis et incidents

La loi de Fresnel étudiée permet de calculer la direction du rayon réfléchi. L'expression vue précédemment pour son calcul peut être utilisée ici. En pratique, on préfère amener l'axe Z selon la direction du rayon. Ceci dans le but de rendre plus simple le calcul de la direction de réfraction [FOLE 90].

IV.8.7 Le calcul de l'intensité lumineuse

Lorsque l'arborescence est créée, elle est traversée en appliquant une équation à chaque noeud pour calculer l'intensité. Par le caractère récursif de cet algorithme, l'intensité pour le noeud courant est obtenue quand tous les sous-noeuds ont été évalués. L'équation appliquée à chaque noeud pour calculer l'intensité peut être différente de celles vues auparavant. On utilise généralement l'équation de Whitted [WHIT 80] ou celle de Hall et Greenberg [HALL 83] qui donne de bons résultats, mais qui est cependant très complexe.

Le modèle de Whitted correspond à celui de Phong, mais adapté au lancer de rayons. On retrouve ainsi une composante correspondant à la lumière ambiante I_{Ambiante} , une composante représentative de la réflexion diffuse I_{Diffuse} , de la réflexion spéculaire $I_{\text{Spéculaire}}$ et, en plus, une composante d'une intensité lumineuse transmise $I_{\text{Transmise}}$;

$$I = I_{\text{Ambiante}} + I_{\text{Diffuse}} + I_{\text{Spéculaire}} + I_{\text{Transmise}}$$

La composante de l'intensité des rayons transmis I_t est multipliée, comme nous l'avons vu dans le cas diffus et spéculaire, par un coefficient de transmission K_t :

$$I_{\text{Transmise}} = K_t * I_t$$

IV.8.8 Conclusion

L'algorithme du lancer de rayons est extrêmement puissant et permet de produire tous les aspects du réalisme. Il tient également compte des ombres portées. Il suffit en effet de détecter quand un rayon allant d'un objet à la source lumineuse est intercepté par un autre objet.

Cet algorithme est souvent utilisé une fois qu'une scène est modélisée et qu'un rendu réaliste de haute qualité est exigé (comme pour les illustrations II et III à la fin de ce chapitre et extraites de [FOLE 90]). Cependant, cette technique prend un temps calcul assez impressionnant et ne permet donc pas une animation en temps réel.

IV.9 La Radiosité

Une des composantes importantes de la perception du relief est l'existence de différences d'éclairements sur la surface des objets. Celles-ci sont dues à l'énergie lumineuse que les objets reçoivent de sources de lumière, mais également d'autres objets, eux-mêmes éclairés. Ces objets fonctionnent alors tels des sources de lumière étendues secondaires. L'ensemble de ces éclairages indirects produit des effets de pénombre, qui sont très difficiles à rendre par les méthodes classiques et même par le lancer de rayons. La technique dite de "radiosité" apporte une solution globale à ce problème [BRET 88]. Elle est cependant assez complexe, et peut être étudiée dans [COHE 85]. Les illustrations IV et V annexées à la fin de ce chapitre montrent le résultat que l'on peut obtenir à l'aide de cette technique.

IV.10 Conclusion

Les algorithmes d'élimination des parties cachées, les nombreux modèles d'illumination, la simulation de la transparence, de textures, d'ombrage et les techniques plus sophistiquées de lancer de rayons et de radiosité que je viens d'exposer dans ce chapitre ont été conçus dans un but précis ; augmenter le réalisme des images de synthèse.

Les résultats que l'on peut obtenir en employant ces techniques peuvent être analysés dans les illustrations VI à XVII [FOLE 90], qui montrent l'évolution progressive du rendu réaliste d'une scène 3D.



Illustration I : "The Snake". Produit par Alias Research™, modélisé avec le système ALIAS™. Il utilise pour le rendu de la peau et du sol des techniques de "texture mapping" et de textures solides. (G. Miller et R. LeBlanc d'Alias™.).



Illustration II : "1984". Réalisé par lancer de rayons avec une résolution de 4096 x 3550 pixels. On peut noter la simulation des mouvements ("Motion-Blurred reflections") et des zones de pénombres dues à des sources étendues. (T. Porter, Copyright Pixar™ 1984).

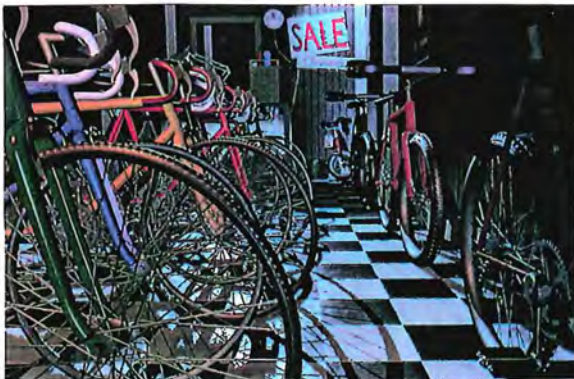


Illustration III "Red's Dream". Réalisé en lancer de rayons. (J. Lasseter, E. Ostby, W. Reeves, et H.B. Siegel, Copyright 1987 Pixar™).

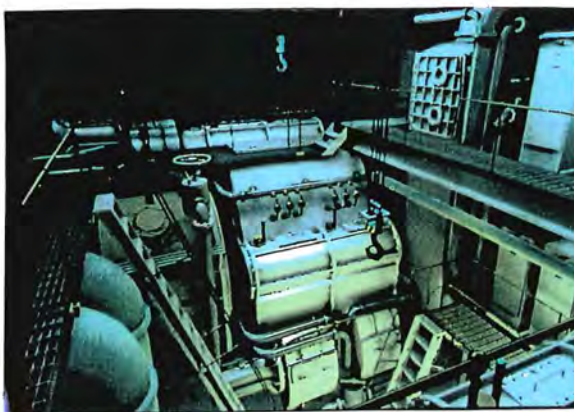


Illustration V : "Boiler Room". Réalisé par un algorithme de Radiosité. (J. Wallace, J. Lin et E. Haines, 1989 Hewlett-Packard™ Co.).

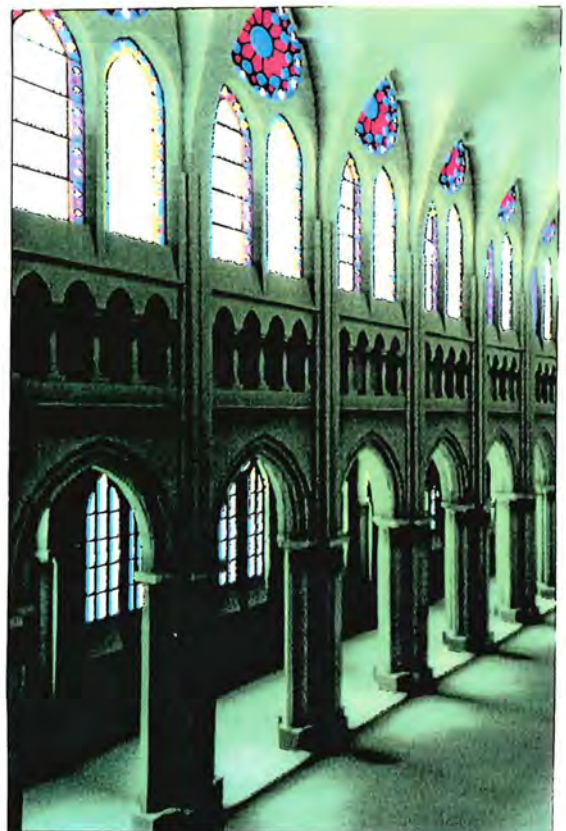


Illustration IV : "Nave of Chartres cathedrale" : Réalisé par un algorithme de Radiosité. Deux travées de la nef, comprenant 9916 polygones, ont été calculées et ensuite recopiées trois fois. (J. Wallace et J. Lin, 1989 Hewlett-Packard™ Co.).

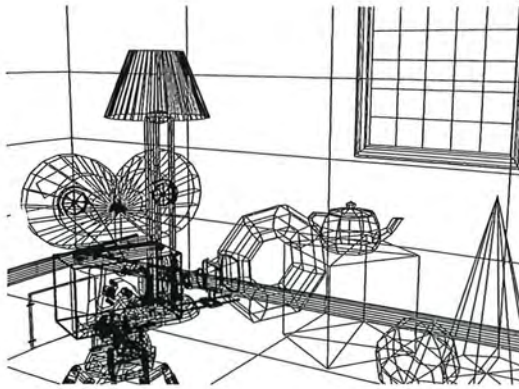


Illustration VI : Détermination d'une projection perspective correspondant au champ de vision d'un observateur particulier.



Illustration VIII : Modèle d'illumination tenant compte simplement de la lumière ambiante.



Illustration X : Modèle d'illumination avec lissage de Gouraud (réflexion diffuse uniquement).

Les illustrations VI à XVIII montrent l'évolution du rendu réaliste d'une scène 3D, modélisée par une approximation polyédrique : "Shutterbug", Copyright 1990, Pixar™. Rendered by T. Williams and H.B. Siegel using Pixar's PhotoRealistic RenderMan™ Software.



Illustration VII : Coloration des arêtes des objets et élimination des parties cachées.



Illustration IX : Modèle d'illumination constante de Lambert (réflexion diffuse uniquement).

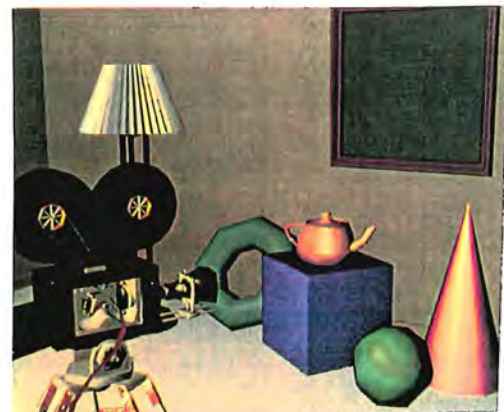


Illustration XI : Modèle d'illumination avec lissage de Gouraud (réflexion diffuse et spéculaire).

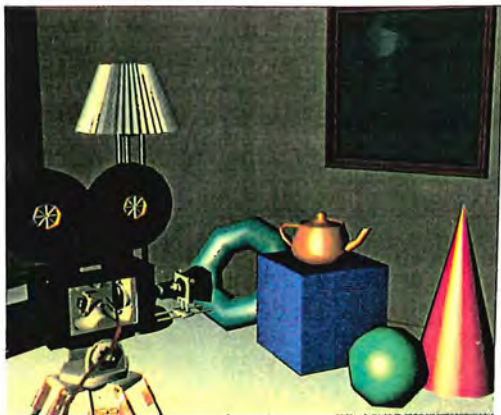


Illustration XII : Modèle d'illumination avec lissage de Phong (réflexion diffuse et spéculaire).

Les illustrations XIII à XVII représentent la même scène que précédemment, mais modélisée par des surfaces paramétriques.

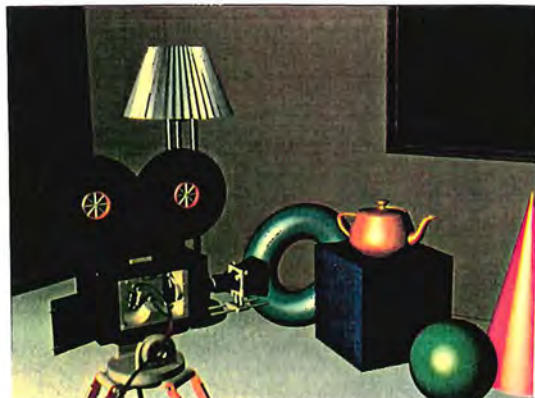


Illustration XIII : Modèle d'illumination totale (réflexion diffuse et spéculaire).



Illustration XIV : Modèle d'illumination totale avec plusieurs sources lumineuses (réflexion diffuse et spéculaire).



Illustration XV : Mapping de textures.



Illustration XVI : Spécification d'une source lumineuse directionnelle et de textures solides.



Illustration XVII : Mapping de textures par réflexion.

<p>V. ED3D : UN EDITEUR HIERARCHIQUE DE SCENES 3D REALISTES</p>
--

V.1 Introduction

Aborder les concepts et les techniques infographiques nécessite dans un premier temps, comme pour toute étude d'un domaine spécifique, la lecture d'ouvrages de référence, tel [FOLE 90]. Mais il est certain que pour bien comprendre les concepts étudiés, une réalisation pratique est de mise.

Le but de ce chapitre est de décrire l'implémentation que j'ai réalisée pendant la période de stage de fin d'études. Je ferai de nombreuses fois référence aux concepts et techniques étudiés dans la partie théorique de ce mémoire (chapitre III et IV). J'invite dès lors le lecteur à s'y rapporter lorsqu'un concept est mal cerné.

Les deux premiers points de ce chapitre décrivent l'environnement de travail dans lequel s'est effectué ce stage, et le dernier présente les concepts et les fonctionnalités du logiciel réalisé.

V.2 Le lieu de stage

Ce stage, d'une durée de cinq mois (de septembre 1990 à janvier 1991), s'est déroulé au laboratoire d'infographie - le LIG - de l'Ecole Polytechnique Fédérale de Lausanne (EPFL). Ce laboratoire, dirigé par Daniel Thalmann, professeur d'infographie à l'EPFL, est récent (1988). Les projets du LIG sont entrepris conjointement avec Nadia Magnenat-Thalmann, professeur à l'Université de Genève. Ils ont pour thème global la synthèse et l'animation de personnages et de décors réalistes, les techniques d'interaction graphique 3D, et la *réalité virtuelle*. Ces thèmes sont dans la continuité des recherches menées depuis une quinzaine d'années au Laboratoire MiraLab de l'Université de Montréal, qui fut dirigé par N. Magnenat-Thalmann et D. Thalmann. Ces derniers sont notamment auteurs de nombreux articles scientifiques et ouvrages consacrés à la synthèse et à l'animation d'images. Pour le grand public et la télévision, N. Magnenat-Thalmann et D. Thalmann sont coréalisateurs de plusieurs films d'animation par ordinateur dont "Vol de Rêve" en 1983 et "Rendez-vous à Montréal" en 1987 [MAGN 87].

Plus spécifiquement, les projets de recherche du LIG sont effectués dans les directions principales suivantes :

- *La simulation et l'animation d'acteurs synthétiques dans leur environnement.*

Cet axe de recherche repose sur des techniques d'intelligence artificielle et de robotique. Il s'agit de modéliser le corps et le visage humain, de simuler la marche

comportementale et l'évitement d'obstacles lors d'une animation, de déterminer la position d'objets en fonction de contraintes, d'étudier le problème de la préhension d'objets, de générer automatiquement des expressions humaines (animation faciale),... On doit tenir compte, dans cet axe de recherche, des lois de la mécanique, de la biomécanique et des aspects comportementaux de l'être humain.

- La visualisation scientifique et les phénomènes naturels.

La recherche en animation s'est surtout consacrée ces dernières années sur le mouvement d'objets rigides. Dans ce type de projet, le but est de développer des techniques d'animation pour déformer des objets soumis à des contraintes physiques. En particulier, on étudie les problèmes d'objets déformables et flexibles, de tissus soumis à des forces (vêtements, drapeau flottant,...), d'objets "mous". On tente de simuler également par ces techniques, le comportement d'objets dans des liquides ou la conduction de chaleur dans les matériaux.

- La simulation de l'apparence des matériaux.

La simulation de matériaux solides, de liquides, et de gaz est un domaine qui prend un essor important. En effet, les super-ordinateurs actuels permettent enfin de calculer la structure de tels matériaux. Cependant, pour visualiser l'apparence de ces matériaux, on doit développer de nouvelles techniques de textures ou de nouveaux modèles stochastiques, basés sur les surfaces fractales ou sur des systèmes de particules. Dans cet axe de recherche, on s'intéresse également à l'affichage de ce type de surfaces par "Ray-Tracing" qui permet un rendu réaliste de grande qualité.

V.3 L'environnement de travail

Le LIG dispose de technologies de pointe en matière de synthèse et d'animation d'images réalistes. Il est équipé principalement ; d'un réseau de stations de travail "**Silicon Graphics™ - POWER Iris**" (Personal Iris, Iris 4D/GT, Iris 4D/GTX, Iris POWERVISION), de périphériques d'entrée et de sortie graphiques sophistiqués tels le "*SpaceBall*" (qui permet de véritablement *manipuler* des objets 3D selon plusieurs degrés de liberté) et le "*DataGlove*" (gant transmettant de façon numérique la position 3D des doigts et des phalanges de la main), et de dispositifs d'impression et de sortie vidéo professionnels SONY™.

Ces stations graphiques professionnelles disposent d'une bibliothèque graphique de haut niveau, la "Graphics Library" - **GL**.

Il est important de mettre en évidence les caractéristiques principales de ces stations POWER Iris et de leur bibliothèque GL. En effet, particulièrement en infographie, le matériel utilisé détermine dans une large mesure les choix de conception et d'implémentation d'un logiciel graphique. Tirer parti de toutes les possibilités offertes en "Hardware" (l'architecture POWER Iris) et en "Software" (GL) par ce type de stations de travail graphique permet d'épargner du temps calcul, nécessaire et si précieux dans la réalisation d'applications graphiques interactives.

V.3.1 La "POWER Iris"¹⁰

A. Description générale

La station Silicon Graphics™ POWER Iris utilise la plupart des techniques infographiques maîtrisées par les constructeurs. Comme la plupart de ses concurrents, comme la station ARDENT Titan [BORD 89], la MEGATEK Sigma 70 [MEGA 89], et la STELLAR gs2000 [APGA 88], la POWER Iris est une station de travail graphique à haute performance. Elle est conçue pour combiner le traitement d'applications générales et d'applications graphiques 2D et 3D à haute vitesse.

L'Iris est équipée, outre le clavier et la souris optique, d'un écran matriciel) rafraîchi ("raster scan display") de haute qualité. Ce type d'écran permet, par un balayage ligne par ligne de l'écran à une fréquence régulière, de créer une image point par point à partir d'une mémoire de rafraîchissement. Cette mémoire est constituée d'un ensemble de plans mémoires, appelés les "*BitPlanes*". Un plan mémoire, organisé en matrice de points, contient exactement un bit d'information pour chaque pixel de l'écran. Puisque l'écran de l'Iris comporte 1024 pixels en hauteur et 1280 pixels en largeur, chaque plan mémoire est constitué de $1280 \times 1024 = 1310720$ bits d'information. Le nombre de plans mémoires varie selon le modèle, le maximum étant 268 ! L'Iris 4D/GT dispose, par exemple, de 96 plans mémoires. Ainsi, 96 bits d'information peuvent être enregistrés pour chaque pixel de l'écran. Ces bits ne constituent pas seulement les informations sur la couleur des points (sous la forme de triplets de valeurs RGB), mais aussi des informations sur la profondeur (nécessaire à la technique du Z-Buffer), et d'autres informations utiles, comme par exemple, un facteur de transparence (la valeur *alpha*).

Pour information, décrivons plus en détail le principe de l'écran de l'Iris... Il est composé d'un tube cathodique couleur à masque ("CRT display"). Ce dernier possède trois canons à électrons qui balayent l'écran entier soixante fois par seconde. Durant ce balayage,

¹⁰Références bibliographiques principales [AKEL 88,89] et [FOLE 90].

chaque canon atteint directement chacun des 1310720 pixels de l'écran pendant un temps très court . Ces canons envoient des électrons qui heurtent l'écran. Chaque pixel de l'écran est composé de trois phosphores différents qui rayonnent soit le rouge, soit le vert ou soit le bleu, correspondant aux trois couleurs primaires auxquelles l'oeil humain est sensible. Un canon active seulement le phosphore rouge, un autre seulement le vert, et le dernier seulement le bleu. Comme chaque canon balaye tous les pixels de l'écran, le nombre d'électrons envoyés (donc l'intensité du rayonnement) est modifié au niveau du pixel.

Considérons seulement le canon à électrons pour le rouge. Si aucun électron n'est envoyé par ce canon à un pixel de l'écran, le phosphore de ce dernier ne rayonnera pas du tout, et ce pixel apparaîtra noir. Si ce canon est activé avec son intensité maximale, le phosphore du pixel rayonnera un rouge vif. Pour les intensités intermédiaires, la couleur du pixel variera entre les deux, du noir au rouge vif. Le principe est identique pour les deux autres canons, excepté que la couleur variera du noir au vert vif, et du noir au bleu vif. La couleur que l'oeil perçoit en un pixel représente en fait la combinaison des intensités du rayonnement des trois phosphores. Les différentes valeurs du triplet des intensités (Rouge, Vert, et Bleu, ou en anglais RGB) pour un pixel produiront la *palette* de couleurs disponibles.

Chacun des canons de couleurs de l'Iris peut prendre 256 valeurs d'intensités différentes. Une intensité est codée sur l'Iris par un vecteur de 8 bits et exige donc 24 (3×8) bits de données pour chaque pixel de l'écran. La valeur 0 correspond à éteindre le canon et 255 à l'intensité maximale. Les intensités des canons Rouge, Vert et Bleu détermineront la couleur du pixel touché. Ces valeurs sont exprimées par un triplet de trois nombres situés entre 0 et 255. Ainsi, (0,0,0) représente le noir, (255,0,0) le rouge vif, et (255,255,255) le blanc. Puisque chacun des canons à 256 niveaux d'intensité, il existe $256 * 256 * 256 = 16777216$ différentes couleurs disponibles.

B. L'architecture POWER Iris 4D-GTX

La POWER Iris 4D-GTX comporte une unité de traitement puissante composée de quatre multiprocesseurs couplés étroitement et partageant un bus de mémoire unique. Son sous-système graphique peut visualiser simultanément, pour la version POWERVISION jusqu'à un million de polygones illuminés par seconde, avec lissage de Gouraud et élimination des parties cachées par Z-Buffer ! La POWER Iris est dans la continuité de la tradition de Silicon Graphics dans l'usage intensif de circuits spécialisés à large échelle ("VLSI"). L'architecture de la POWER Iris, schématisée à la figure V.1 extraite de [FOLE 90], est composé de cinq sous-systèmes majeurs.

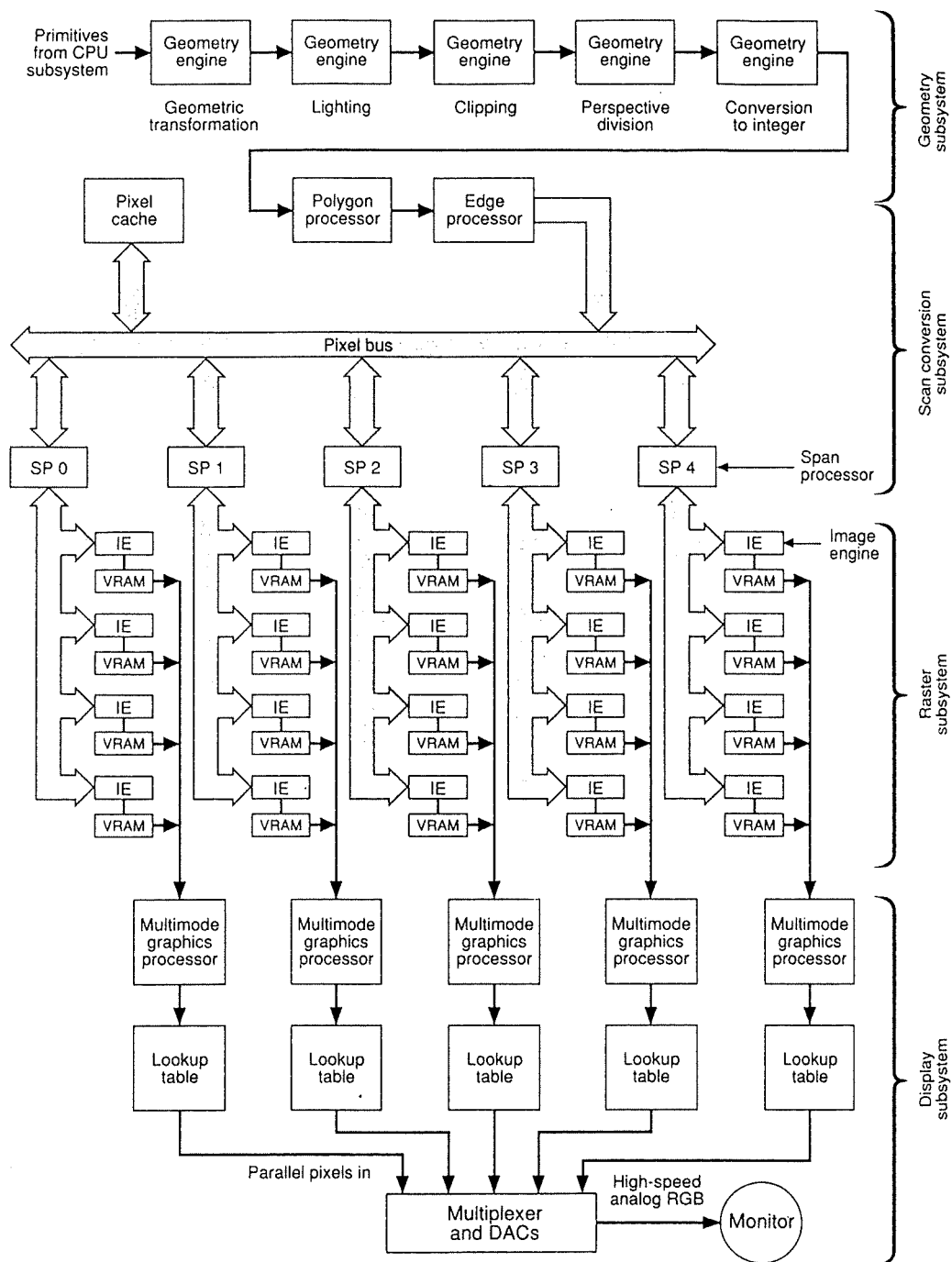


Figure V.1 : L'architecture POWER Iris 4D-GTX.

1) Le sous-système de traitement - "CPU Subsystem".

Il exécute l'application et parcourt la base de données (la structure de données à afficher). Il est composé de quatre multiprocesseurs couplés, symétriques, et partageant la même mémoire. Le matériel fournit une synchronisation rapide entre les processeurs.

2) Le sous-système de géométrie - "Geometry Engine".

Il effectue les transformations géométriques, détermine le découpage nécessaire des primitives graphiques pour correspondre aux coordonnées de l'écran, et offre des primitives d'illumination. Il est composé de cinq processeurs disposés en série ("pipeline architecture"). Chacun de ces processeurs, appelé moteur de géométrie - "Geometry Engine" - ou GE, contient une entrée FIFO, un contrôleur, et une unité de calcul en virgule flottante d'une puissance de 20 MFlops : le Weitek 3332.

3) Le sous-système de conversion point à point - "Scan-Conversion Subsystem".

Il convertit les primitives de dessins (points, lignes, et polygones) en matrice de points de l'écran (en "BitMap"). Il utilise un algorithme de conversion point à point appelé "Rasterization Algorithm" [FOLE 90].

4) Le sous-système de visualisation - "Raster subsystem".

Il traite le problème de l'élimination des parties cachées. A partir des données des points générés par le sous-système de conversion point à point, il met à jour l'image et les "BitPlanes" correspondant au Z-Buffer dans la mémoire d'écran, en utilisant le résultat de la comparaison des valeurs des coordonnées Z. Ce sous-système est composé de vingt moteurs d'image ("Image Engine") travaillant en parallèle, chacun étant responsable pour un vingtième des points de l'écran, disposés en tableaux de quatre sur cinq points.

5) Le sous-système d'affichage - "Display subsystem".

Il contient cinq processeurs graphiques multimode, appelés "MGP". Chacun assigne un cinquième des colonnes de l'affichage. Ces processeurs lisent les données de l'image dans la mémoire d'écran, et les envoient dans les convertisseurs digitaux-analogiques pour l'affichage.

C. Conclusion

Cette architecture matérielle, mise au point par Silicon Graphics™, fournit un haut degré de performance pour le rendu réaliste en temps réel. Elle peut être exploitée pleinement par les procédures offertes par la bibliothèque graphique de l'Iris, nommée GL.

V.3.2 La "GL" de l'Iris 4D - GT¹¹

Je me limiterai ici à décrire les possibilités principales de cette bibliothèque développée par Silicon Graphics [SGUS 88], et utilisée pour le développement du logiciel réalisé lors de mon stage.

A. Les primitives graphiques

GL offre de nombreuses primitives de dessin de figures géométriques : points, lignes, polygones, rectangles, cercles, arcs de cercles, réseaux de triangles, courbes cubiques paramétriques (Bézier et B-Spline) en 2D ou en 3D. Il est important de réaliser que les figures 3D sont réellement considérées comme telles par l'Iris : elles sont ainsi représentées par une description géométrique 3D, à savoir par un ensemble de coordonnées (x, y, z). Ces figures géométriques peuvent subir des transformations géométriques (rotations, translations, changement d'échelle). L'ensemble de ces figures géométriques définit une *scène* à visualiser. Il est alors nécessaire de définir le point de vue de l'observateur pour obtenir ainsi une projection géométrique (perspective ou orthogonale) de la scène créée.

B. La coloration et le "Shading" de Gouraud

Les routines GL permettent de créer des figures géométriques avec une coloration uniforme (un polygone bleu, ou une ligne rouge), ils sont appelés "flat shaded polygons". Or, le modèle polygonal est souvent utilisé pour approcher une surface ou une courbe. Si c'est le cas, les couleurs ont tendance normalement à varier de manière continue sur toute la surface. Une coloration uniforme sur une surface approchée par des polygones a donc un faible degré de réalisme.

Une première solution pour éviter ce désagrément, et supprimer l'apparence de ces polygones sur la surface est d'utiliser un très grand nombre de petits polygones pour approcher la surface. Mais, un moyen beaucoup plus simple est de faire varier la couleur sur la surface des polygones, c'est-à-dire réaliser un lissage (ou couramment appelé un "Shading") de Gouraud. Il suffit de calculer la couleur correcte pour la surface réelle à chaque sommet du polygone approximatif, et le matériel fera varier la couleur de celui-ci en se basant sur ces valeurs. L'Iris 4D/GT réalise cette opération très rapidement. Les modèles d'illumination calculent alors seulement les couleurs aux sommets du polygone par interpolation.

¹¹Référence bibliographie : [SGUS 88]

Prenons un exemple, afin de compléter la description théorique de cette technique présentée dans le chapitre IV.

On suppose un côté d'un polygone de six points de longueur avec la couleur (0,20,100) à une extrémité et de (75,60,50) à l'autre ; ces triplets représentant les valeurs RGB définissant la couleur. Les six points seront coloriés de la manière suivante : (0,20,100), (15,28,90), (30,36,80), (45,44,70), (69,52,60), et (75,60,50). Notons que chaque composant de la couleur varie doucement du premier point jusqu'au dernier. Le composant Rouge augmente de cinq à chaque fois, le composant Vert augmente de huit, et le composant Bleu diminue de dix pour chaque point. Dans ce cas, la différence dans la couleur des points est entière. Habituellement, ce n'est pas le cas mais l'approximation est réalisée de façon aussi précise que possible.

Après que les couleurs des points sur les côtés du polygone ont été déterminées, le même processus est utilisé pour trouver les couleurs des points situés à l'intérieur du polygone. La figure V.2 montre le résultat d'un triangle dont les sommets ont les couleurs (0,20,100), (75,60,50), et (0,0,0).

(0,20,100)					
(15,28,90)	(0,16,80)				
(30,36,80)	(15,24,70)	(0,12,60)			
(45,44,70)	(30,32,60)	(12,20,50)	(0,8,40)		
(60,52,60)	(45,40,50)	(30,28,40)	(15,16,30)	(0,4,20)	
(75,60,50)	(60,48,40)	(45,36,30)	(30,24,20)	(15,12,10)	(0,0,0)

Figure V.2 : Un triangle illuminé par un Shading de Gouraud

C. L'animation et le "Double Buffering"

Une fois que les informations concernant les couleurs des points à afficher sont écrites dans la mémoire d'écran, le résultat est immédiatement visible à l'écran. Pour des images statiques, c'est acceptable, mais cela ne convient pas pour une animation. On observe en effet, lorsque les images sont quelque peu complexes, un léger clignotement entre les différentes séquences de l'animation.

Pour une animation "douce", il est préférable d'afficher une image complètement dessinée pour une certaine période de temps (par exemple, environ un sixième de secondes), et ensuite l'image suivante, également complètement dessinée, durant la période

de temps suivante, et ainsi de suite...

La technique du doublement mémoire, dit du "double buffering", de la GL offre cette possibilité. Les plans mémoires standard de l'Iris sont alors divisés en deux parties : une des deux est affichée, et on dessine simultanément dans l'autre. Quand le dessin est complet, on *permut*e les deux mémoires, et la mémoire auparavant invisible et contenant l'image suivante devient visible, et la mémoire auparavant visible devient invisible et est disponible pour le dessin de l'image suivante. La mémoire visible est appelée la mémoire d'avant-plan, le "front buffer", et la mémoire invisible et destinée au dessin est appelée la mémoire d'arrière-plan, le "back buffer".

D. Les transformations de coordonnées

Une fois que l'on est capable de construire des objets 3D, il est intéressant de pouvoir les bouger relativement à d'autres objets ou relativement à l'observateur, de leur faire subir des rotations, des translations ou des changements d'échelle. On doit également être capable de modifier le point de vue de l'observateur, son champ de vision, et son orientation. Plusieurs routines GL permettent de réaliser ces opérations géométriques 3D, correspondant à des transformations de coordonnées.

Les opérations 3D peuvent être subdivisées en trois types : les transformations de modélisation, de visualisation et de projection, matérialisées par des matrices.

L'étude de ces opérations de transformations a déjà été abordée dans la partie théorique de ce mémoire (chapitre III). Il serait cependant utile de présenter succinctement comment ces transformations sont réalisées concrètement par l'Iris. Mais avant cela, quelques notions sur les systèmes de coordonnées définis par l'Iris sont nécessaires.

1) Les systèmes de coordonnées

Il existe trois systèmes de coordonnées nécessaires à l'affichage d'une scène, bien que l'on puisse penser se munir de plusieurs autres systèmes de coordonnées dans certaines situations, pour des raisons de facilité.

Le premier est un système 3D qui définit les coordonnées par des triplets de nombres réels. Il est appelé le *système de coordonnées de référence* (ou *du monde*), dans lequel tous les objets de la scène sont décrits. Il n'existe pas, a priori, de limite à la valeur de ces coordonnées (autre que la taille maximale d'une valeur réelle sur la machine).

Le second système est appelé le *système de coordonnées normalisées*. Il est également défini par des triplets de nombres réels, mais dont les valeurs x, y, et z sont

comprises entre -1.0 et 1.0. Ce "cube" 3D est très intéressant pour les procédures de découpage. En effet, après les transformations des coordonnées du monde dans le système de coordonnées normalisées, les procédures (câblées) de découpage éliminent toute géométrie dont les coordonnées sont à l'extérieur de l'intervalle $[-1.0; 1.0]$.

Si l'on dessine dans une fenêtre placée de manière aléatoire sur l'écran, le pixel situé dans le coin inférieur gauche de celle-ci possède comme coordonnée "écran" (0,0). Les coordonnées normalisées sont donc ensuite transformées directement dans le troisième système de coordonnées, habituellement appelé le *système de coordonnées écran*. Les coordonnées *écran* sont des nombres entiers. On peut remarquer que le passage des coordonnées normalisées en coordonnées *écran* peut nécessiter une certaine approximation et donc occasionner une perte de précision. On pense généralement que les coordonnées *écran* sont représentées en 2D. Or, les trois coordonnées normalisées x , y et z sont réduites, et il existe véritablement une coordonnée z *écran* qui est très utile pour maintes opérations, comme par exemple l'élimination des parties cachées.

2) Les transformations de "Projection"

GL offre plusieurs primitives qui permettent de réaliser une transformation de projection. Elles fournissent donc le moyen de créer de véritables *caméras virtuelles*.

Il est possible de simuler une projection perspective, qui correspond à une vue obtenue avec un oeil situé à l'origine et regardant le long de l'axe Z négatif.

Le volume de vision de ce type de projection forme virtuellement une pyramide dont le sommet correspond à l'oeil de l'observateur. Tout ce qui se trouve à l'extérieur de cette pyramide n'apparaîtra pas. Par conséquent, les quatre plans qui déterminent ce volume de vision de l'observateur détermineront le découpage (appelé le "le clipping") à opérer sur la scène. Ils sont appelés les plans "*left*", "*right*", "*top*", et "*bottom*". Le matériel prévoit également deux autres plans de découpage qui éliminent tout ce qui se trouve trop près et trop loin de l'oeil ; ils sont appelés les plans "*near*" et "*far*". Ainsi, pour une visualisation en perspective, la région du "monde" qui est visible sur écran ressemble à une pyramide égyptienne dont le sommet est sectionné par le plan de clipping "*near*" (voir figure V.3, extraite de [SGUS 88]). Le nom technique de cette pyramide est un "*frustum*".

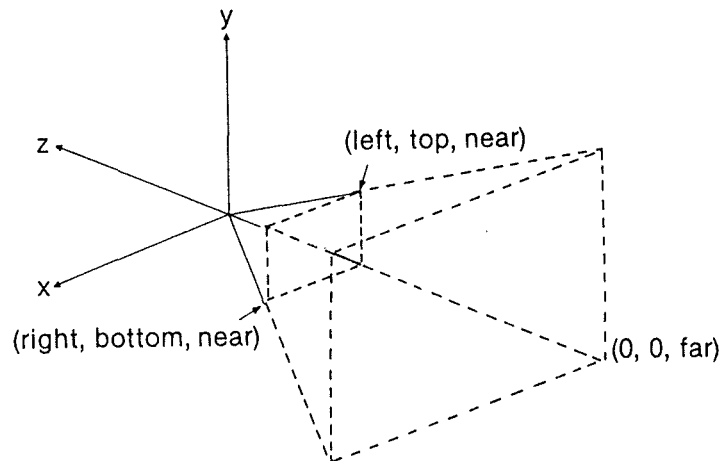


Figure V.3 : Le volume de vision d'une projection perspective.

D'un point de vue mathématique, la procédure qui réalise la projection perspective doit faire correspondre le volume 3D défini par le "*frustum*" spécifié en coordonnées normalisées. Tout point situé à l'extérieur de ce "*frustum*" sera considéré comme un point extérieur au "cube" 3D normalisé (par conséquent, au moins une de ses coordonnées sera supérieure à 1.0 ou inférieure à -1.0). Comme vu précédemment, le matériel élimine alors toute géométrie externe au cube 3D et transforme ensuite les coordonnées normalisées en coordonnées écran pour la fenêtre courante à l'écran.

On peut également définir une projection orthogonale plutôt que perspective. Le volume de vision de cette projection est matérialisé dans ce cas à un parallélépipède rectangle, appelé "*rectangular viewing frustum*". Ce type de projection correspond à une projection perspective où l'oeil de l'observateur est situé à l'infini. D'un point de vue mathématique, toute la géométrie située à l'extérieur de ce volume de vision est découpée et la géométrie située à l'intérieur de ce volume est alors projetée parallèlement à l'axe Z sur le plan "*far*" parallèle au plan XY (voir figure V.4, extraite de [SGUS 88]).

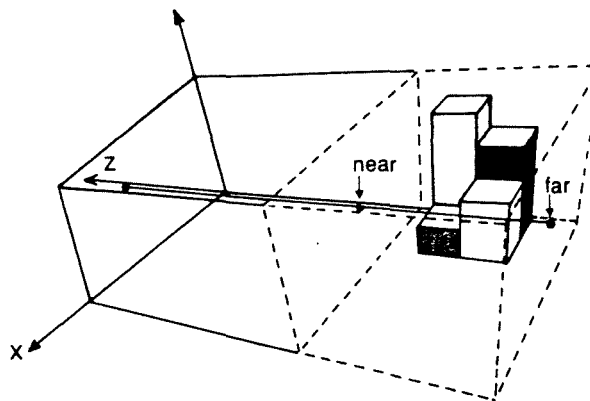


Figure V.4 : Le volume de vision d'une projection orthogonale.

3) Les transformations de "viewing"

Les transformations de projection de GL vues précédemment supposent que l'oeil de l'observateur regarde le long de l'axe Z négatif et, pour la projection perspective, que cet oeil est situé à l'origine. Dans le cas de projection orthogonale, il n'est pas sensé de parler de la position exacte de cet oeil, mais seulement en terme de direction de vision.

Les transformations de visualisation permettent de spécifier la position de l'oeil de l'observateur dans le système de coordonnées du monde, et de spécifier la direction selon laquelle il regarde. Dans le cas d'une projection orthogonale, seule la direction de vision est prise en compte. D'un point de vue mathématique, ces transformations de visualisation transforment, par une succession de rotations et de translations, la position de l'oeil de manière à le transporter à l'origine, et qu'il regarde le long de l'axe Z négatif pour pouvoir appliquer les transformations de projection.

4) Les transformations de "modeling"

Chaque objet de la scène est défini dans son propre système de coordonnées. On peut dès lors manipuler un objet entier en utilisant des transformations de "Modeling" telles : les rotations, les translations et les changements d'échelle. On peut évidemment combiner ces transformations entre elles ou avec des routines de dessin, pour exprimer des interrelations entre différentes parties d'un objet complexe.

Il est important de souligner que la combinaison de telles transformations sur un objet n'est pas commutative... En effet, la figure V.5 montre deux séquences différentes d'une translation et d'une rotation sur un objet simple.

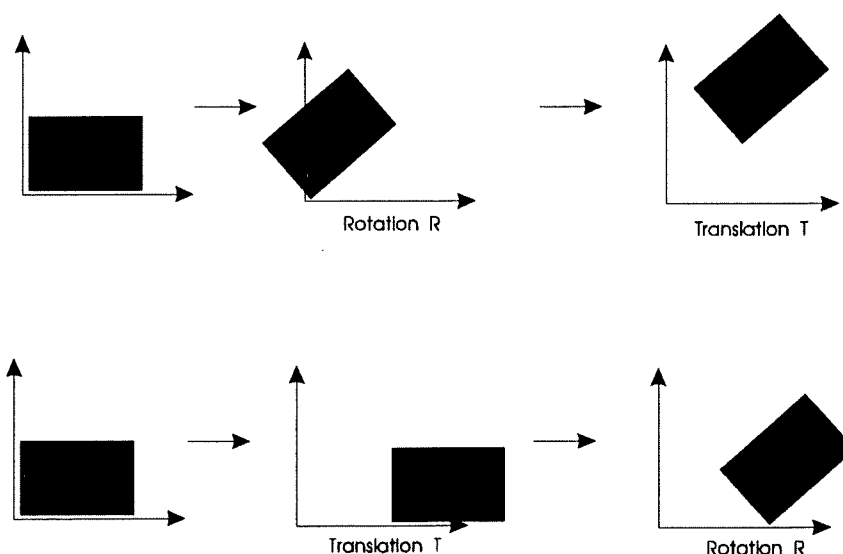


Figure V.5 : La non-commutativité des transformations géométriques.

E. Les opérations matricielles hiérarchiques

Une pile mémoire de type "Matrice" est à disposition pour les opérations matricielles. L'utilité de cette pile matricielle peut se justifier par un simple exemple ...

Un dessin est souvent composé de plusieurs copies de dessins plus simples, qui sont eux-mêmes composés de dessins plus simples encore... Par exemple, pour un vélo, on peut ne disposer que d'une unique procédure qui sera appelée deux fois pour dessiner une roue disposée de manière appropriée. Ceci de même pour le dessin des rayons des deux roues. On peut également exiger de dessiner ce vélo plusieurs fois dans la scène à visualiser.

Supposons que le vélo soit décrit dans un système de coordonnées où l'origine correspond au braquet du vélo, et la roue dans un système de coordonnées où l'origine se trouve au centre de celle-ci. Il faut dès lors, après avoir dessiné le vélo, *déplacer* l'origine du système de coordonnées de manière à dessiner à la bonne place la première roue, elle-même définie dans son propre système de coordonnées. Puis revenir en arrière pour dessiner une nouvelle instance de roue.

Examinons ceci d'un point de vue mathématique. Notons M la matrice de "Modeling" qui décrit la position du vélo, et S et T les matrices de "Modeling" **relatives à M** pour se déplacer respectivement vers la position de la roue avant et arrière. Il faudra, par conséquent, pour dessiner la roue avant utiliser la matrice de transformation $S*M$ et la matrice $T*M$ pour dessiner la roue arrière.

Cette opération peut être réalisée avec la pile matricielle (jusqu'à 32 matrices). A tout moment d'un dessin d'une scène, il existe une matrice de transformation courante qui se situe au sommet de la pile matricielle, et qui est la résultante de toutes les compositions des transformations effectuées jusqu'alors. Tout sommet à afficher est auparavant transformé par la matrice courante, située au sommet de la pile, et qui représente la transformation attendue.

Deux routines GL, très importantes, permettent de réaliser ces opérations sur cette pile matricielle : ***PushMatrix*** et ***PopMatrix***.

La première routine *pousse* la pile de matrices vers le bas et recopie la matrice courante (par exemple, M) au sommet ; il reste donc après l'application de cette procédure deux copies de M . Lorsque l'on désire effectuer une transformation (par exemple, la translation S), on doit avoir $S*M$ au sommet de la pile pour obtenir la bonne transformation. Pour ce faire, on prémultiplie la matrice située au sommet de la pile (M) par S . On a donc la matrice $S*M$ au sommet et M juste en-dessous. Dans l'exemple du

vélo, la roue est dessinée en utilisant S*M comme matrice de "modeling".

PopMatrix permet ensuite de *dépiler* S*M, et de laisser ainsi de nouveau M au sommet de la pile, pour continuer le dessin...

Il est possible de donner un algorithme en pseudo-code du dessin complet du vélo :

```
/* M au sommet de la pile */  
  
PushMatrix ( ).                /* 2 matrices M sur la pile */  
  
Translator (matrice de transformation S) vers la position de la roue avant,  
c'est-à-dire multiplier M par S. /* S*M, puis M sur la pile */  
  
Dessiner une roue.  
  
PopMatrix ( ).                /* M sur la pile */  
  
PushMatrix ( ).                /* 2 matrices M sur la pile */  
  
Translator (matrice de transformation T) vers la position de la roue arrière,  
c'est-à-dire multiplier M par T. /* T*M, puis M sur la pile */  
  
Dessiner une roue.  
  
PopMatrix ( ).                /* M sur la pile */  
  
Continuer le dessin du vélo.
```

F. L'élimination des parties cachées

Par défaut, l'Iris ne réalise pas l'élimination des parties cachées ; la scène est affichée à l'écran dans l'ordre où les figures qui la composent sont dessinées. Or, pour la plupart des scènes 3D, il est important de n'afficher que les surfaces qui se trouvent le plus près de l'oeil, tout du moins pour les objets opaques. Toutes les autres surfaces seront cachées par ces parties plus proches de l'oeil.

L'élimination des parties cachées peut cependant être réalisée sur l'Iris en utilisant la technique du Z-Buffer. Ce dernier est composé d'une mémoire de nombres de 24 bits : un bit pour chaque pixel de l'écran. Pour rappel, la technique est la suivante : au début, on initialise tous les nombres de cette mémoire à la plus grande valeur possible. Ensuite, lorsqu'une primitive géométrique (un polygone, une ligne, un point ou un caractère) est à afficher, leurs coordonnées x et y sont calculées de la manière habituelle, et avant que le pixel ne soit colorié dans la couleur de la primitive, sa coordonnée z est calculée. Cette valeur z correspond en fait à la distance de ce pixel par rapport à l'oeil de l'observateur. On compare ensuite cette valeur à la valeur correspondante pour ce pixel dans le Z-Buffer. Si la valeur Z calculée est inférieure à la valeur présente dans le Z-Buffer, le point est colorié et

on place cette nouvelle valeur dans le Z-Buffer. Ainsi, à tout moment, les valeurs présentes dans le Z-Buffer représentent les distances des points qui sont les plus proches de l'oeil de l'observateur. Les valeurs dans les autres plans mémoires représentent la couleur de ces points, et représentent donc la couleur à afficher.

G. L'illumination

L'Iris peut exécuter rapidement certains calculs d'illumination, nécessaires pour augmenter le degré de réalisme de la géométrie affichée à l'écran.

Les modèles d'illumination ont été étudiés en détail dans le chapitre IV, mais il serait cependant utile de rappeler le fonctionnement général de ceux-ci, afin qu'une compréhension globale par le lecteur de ces mécanismes soit assurée.

1) Rappel des principes de base

Lorsque l'on observe un objet dans le monde réel, la couleur de celui-ci (celle perçue par l'oeil) dépend de plusieurs paramètres :

- 1) la couleur, la position et la direction des sources de lumière qui illuminent cet objet,
- 2) la couleur et les propriétés de la surface de l'objet lui-même,
- 3) la position et la direction de vision de l'observateur.

Considérons un cylindre de surface opaque grise illuminé par une source de lumière blanche. Les objets de surface rugueuse ont tendance à réfléchir la lumière de manière égale dans toutes les directions. De telles réflexions sont appelées *réflexions diffuses*. La couleur et l'intensité de la lumière réfléchie de manière diffuse en un point de la surface du cylindre dépend de trois facteurs :

- 1) la couleur de la source de lumière,
- 2) la valeur de l'angle entre la direction de la source de lumière et l'orientation de la surface du cylindre en ce point (qui est spécifiée par le vecteur normal à la surface en ce point),
- 3) les caractéristiques de réflexion de la surface du cylindre.

Puisque le vecteur normal est différent en chaque point du cylindre, le modèle d'illumination doit donc produire une valeur différente pour tous les points du cylindre. Il est important de souligner que ce vecteur normal doit être normalisé (sa longueur doit être égale à 1).

Comme mentionné précédemment, les surfaces des objets courbes sont approchées la plupart du temps par un réseau de facettes planes. Le degré de réalisme d'un objet sera ainsi

fonction du nombre de sommets (et donc du nombre de facettes) qui le définissent. Comme un calcul d'illumination est exécuté pour tous les sommets de l'objet, plus ce nombre est élevé, plus le degré de réalisme sera élevé.

En plus de la lumière réfléchiée par l'objet et qui provient de sources lumineuses présentes dans la scène, il faut également considérer une lumière ambiante. Dans la réalité, la lumière ambiante est la lumière qui se réfléchit sur une ou plusieurs surfaces dans la scène avant d'arriver sur la surface en question. On considère souvent que la lumière ambiante provient de toutes les directions et qu'elle est réfléchiée uniformément dans toutes les directions par la surface en question. Par conséquent, la couleur et l'intensité de la lumière ambiante réfléchiée sont fonction à la fois du niveau de lumière ambiante présente dans la scène et des propriétés physiques de réflexion de la surface de l'objet. Ce n'est plus fonction de l'angle formé par la source lumineuse et le vecteur normal à la surface.

A l'instar des surfaces rugueuses qui réfléchissent la lumière dans toutes les directions, les surfaces lisses ont tendance à réfléchir la lumière de façon directionnelle. Cette réflexion directionnelle cause les reflets caractéristiques des surfaces métalliques, et est appelé *réflexion spéculaire*. De plus, ce type de réflexion dépend non seulement de l'angle formé par le vecteur normal à la surface et par la direction de la source de lumière, mais également de la direction de vision de l'observateur. Quand cette direction de vision coïncide avec la direction des rayons réfléchis, l'observateur perçoit un reflet spéculaire, un *éclat*. Sur une surface parfaitement lisse tel un miroir, la direction des rayons de lumière réfléchis est égale à l'angle d'incidence (c'est-à-dire l'angle formé par le vecteur directeur de la source lumineuse et le vecteur normal à la surface). Sur les surfaces moins lisses, une dispersion de la lumière réfléchiée se produit. Aussi, comme le représente la figure V.6, l'angle de réflexion des rayons réfléchis n'est pas tout à fait égal à l'angle d'incidence.

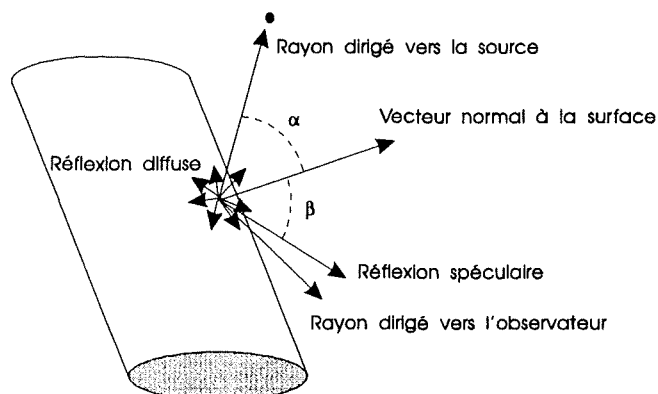


Figure V.6 : La réflexion diffuse et spéculaire.

2) Les calculs d'illumination de GL

Pour effectuer ces calculs d'illumination sur les sommets de tous les objets d'une scène, il est donc nécessaire de fournir à GL toute une série d'informations.

- Tout d'abord, la liste des "normales" à la surface pour tous les sommets des objets de la scène.

- Les propriétés des matériaux des différents objets à simuler ; on spécifie pour chaque matériau un coefficient de réflexion de la lumière ambiante, diffuse et spéculaire, un coefficient d'émissivité qui permet de simuler l'apparence d'une source lumineuse, un coefficient de luisance qui permet de spécifier le degré de concentration des "éclats" et un coefficient de transparence.

- La définition des sources lumineuses présentes dans la scène ; on spécifie le type, la couleur, la position ou la direction (dans le cas de source de lumière directionnelle), une composante ambiante, un coefficient d'atténuation. Les calculs d'illumination sont simplifiés dans le cas d'une unique source de lumière infinie, vu que l'on dispose d'un vecteur directeur constant et que l'on ne doit donc pas le recalculer à chaque fois.

- Le modèle d'illumination à appliquer sur toute la scène ; on spécifie la lumière ambiante, un coefficient d'atténuation, et si l'observateur est situé à l'infini ou localement. Les calculs sont en effet simplifiés dans le premier cas, vu que le vecteur dirigé vers l'oeil de l'observateur ne doit pas être recalculé à chaque fois.

On peut percevoir une nette différence visuelle si l'on spécifie un observateur et/ou une source de lumière situé localement dans le cas de l'affichage d'une figure géométrique qui est constituée de surfaces planes. Le vecteur normal à ce type de surface est bien évidemment constant. Par conséquent, si l'on utilise des sources de lumières infinies et un observateur situé également à l'infini, les angles entre le vecteur normal, le vecteur de vision (en direction de l'oeil de l'observateur), et le vecteur en direction des différentes sources de lumières infinies seront constants, ce qui provoquera une couleur uniforme sur toute la surface. En spécifiant un observateur et/ou une source lumineuse situé localement dans la scène, les vecteurs de vision et/ou de direction de la source lumineuse locale changeront de valeur sur la surface, tout comme la couleur résultante.

H. Le principe d'affichage

L'affichage d'un point sur écran est réalisé par l'Iris comme suit :

1) Les coordonnées de ce point sont transformées dans leur propre système de coordonnées "monde" par la matrice de "Modeling", qui oriente, positionne et détermine le facteur d'échelle de l'objet comprenant ce point.

2) Les coordonnées résultantes sont ensuite transformées par la matrice de "Viewing", qui spécifie la position de l'observateur. Le but est de *transporter* l'oeil de l'observateur à l'origine et de manière à ce qu'il regarde en direction de l'axe Z négatif. On modifie de la sorte la position de ce point pour qu'il corresponde à l'endroit prévu par le champ de vision.

3) Les coordonnées sont finalement modifiées par la matrice de projection qui va permettre d'afficher l'objet avec élimination des parties cachées en utilisant le Z-Buffer.

Considérons un sommet d'une figure (en fait matérialisé par un vecteur colonne x), qui a une matrice de "modeling" M , une matrice de "viewing" V , et une matrice de projection P . Ce point doit donc subir une série de multiplications matricielles ordonnée que l'on peut modéliser mathématiquement par :

$$x \Rightarrow x*M \Rightarrow (x*M)*V \Rightarrow ((x*M)*V)*P \Leftrightarrow x*(M*V*P).$$

La transformation en coordonnées normalisées est accomplie en multipliant les vecteurs 3D des figures de la scène par une matrice qui représente une combinaison des transformations de "modeling", de "viewing" et de "projection". Chacune de ces transformations est représentée par une matrice. Elles peuvent être multipliées entre elles pour obtenir une matrice unique qui aura le même effet que l'application séquentielle de toutes les matrices individuelles. Le matériel concatène donc $M*V*P$ en une seule matrice pour gagner du temps CPU.

Si l'on n'effectue pas de calculs d'illumination, le modèle de transformations géométriques est relativement simple. Les sommets, représentant des positions dans un espace tridimensionnel, sont transformés en coordonnées normalisées, c'est-à-dire dans un cube 3D où les coordonnées X , Y , et Z sont strictement comprises dans l'intervalle $[-1.0, 1.0]$. Ils sont ensuite transformés de nouveau en coordonnées "écran" pour correspondre à la fenêtre physique de l'écran.

Lorsque l'on effectue des calculs d'illumination, cette transformation des vecteurs 3D doit se réaliser en deux étapes. En effet, le modèle de calcul des illuminations nécessite que les vecteurs 3D soient transformés par la matrice de "Modeling" et de "Viewing" mais pas encore "projetés". On doit dès lors maintenir une matrice de modelage/visualisation et une matrice de projection de manière séparée. Tout comme les vecteurs 3D représentant les sommets des figures géométriques, les normales associées à ces sommets doivent être également transformées, mais sont soumises à des lois mathématiques différentes. Les normales sont en fait multipliées par la transposée de l'inverse de la sous-matrice 3x3 supérieure gauche de la matrice de "Modeling/Viewing". Un résumé de la procédure à réaliser lors des calculs d'illumination pour tout vecteur 3D est modélisé à la figure V.7.

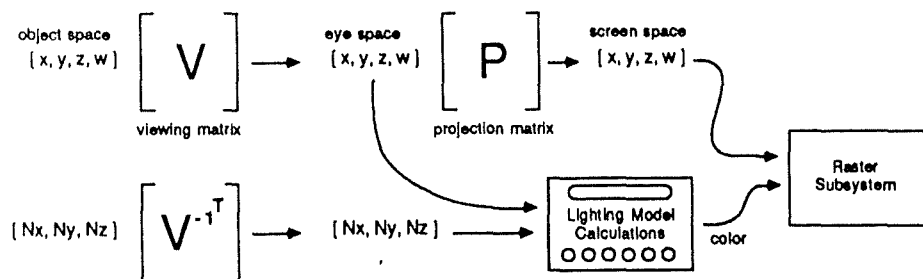


Figure V.7 : Le principe d'affichage avec calculs d'illumination.

I. Temps d'exécution et performance

L'Iris est extrêmement rapide lors de l'affichage de scènes modélisées en *Fil-de-Fer*. La majeure partie du temps d'affichage dépend de la complexité des calculs d'illumination.

L'équation d'illumination est évaluée quand la normale ou la position graphique courante change (c'est-à-dire où l'on va dessiner le prochain sommet ...). Elle dépend également du caractère local ou infini de l'observateur et des sources lumineuses. Si l'observateur ou une des sources lumineuses est local, le calcul de l'illumination est exécuté quand la position graphique change. Quand l'observateur est local, le vecteur du sommet à l'œil de l'observateur est différent pour chaque sommet et doit être calculé pour chaque point. Si l'observateur est situé à l'infini, alors le vecteur de vision est constant pour tous les sommets et le temps de calcul est moindre. Quand une source de lumière est locale, le vecteur du sommet à la source lumineuse est différent pour chaque sommet et doit donc être calculé pour chaque sommet. Si une lumière est à l'infini, alors le vecteur directeur lumineux est constant pour tous les sommets et le temps de calcul est moindre.

La performance est donc supérieure lorsque l'on spécifie dans les calculs d'illumination un observateur et une unique source de lumière situés à l'infini. Le temps

d'exécution augmente légèrement pour toute source de lumière infinie ajoutée au modèle d'illumination. Cependant, l'ajout de sources de lumières locales augmentera le temps d'exécution de façon notable.

Après que le vecteur normal soit transformé par la transposée inverse de la matrice de "Modeling/Viewing", il doit être renormalisé si cette matrice n'est pas orthonormale. Cette renormalisation exige un temps d'exécution additionnel et engendre une moindre performance. Pour rappel, une matrice est orthonormale si tous ses vecteurs colonnes sont de longueur unitaire et orthogonaux aux autres.

V.3.3 Conclusion

L'architecture Iris, combinée avec GL, permet la synthèse et l'animation de formes construites à partir de surfaces polyédriques ou de "*Splines*". Elle fournit également le moyen d'apporter un rendu réaliste en temps réel. Certains algorithmes sont en effet directement câblés, comme par exemple le produit matriciel, le Z-Buffer pour l'élimination des parties cachées, et le lissage de Gouraud pour les calculs d'illumination.

Après avoir décrit l'environnement de travail (humain, matériel et logiciel) dans lequel mon stage s'est déroulé, je présenterai ci-dessous la réalisation pratique effectuée au LIG.

V.4 Présentation du logiciel ED3D

V.4.1 Introduction

Mon stage au LIG s'est déroulé en trois *phases*...

La première eut pour but l'étude des techniques infographiques de base, nécessaires à la synthèse d'images réalistes. Celles-ci ont été exposées dans le chapitre III et IV de ce mémoire.

La seconde période fut dédiée à la familiarisation de la station graphique professionnelle POWER Iris et de sa bibliothèque graphique GL ; toutes deux présentées au début de ce chapitre V.

Afin de concrétiser ces approches théoriques et de maîtriser les possibilités de l'Iris et de GL, une implémentation personnelle s'avérait bénéfique. Après avoir discuté avec les membres du LIG, la spécification d'un **éditeur hiérarchique interactif de scènes tridimensionnelles** fut mise au point. La dernière phase de ce stage de fin d'études eut donc

pour objet la réalisation de ce logiciel, que je présenterai ci-dessous.

V.4.2 Objectifs du logiciel ED3D

ED3D doit permettre l'édition interactive de scènes 3D composées :

- d'objets 3D,
- de sources de lumières ponctuelles ou directionnelles,
- et de caméras virtuelles.

Ces scènes doivent pouvoir être visualisées selon différentes méthodes de rendu réaliste. On propose quatre modes de visualisation privilégiés :

- La représentation par le modèle fil-de-fer.
- La représentation avec élimination des faces cachées en couleurs "pleines".
- Le modèle d'illumination constante de Lambert.
- Le modèle d'illumination avec lissage de Gouraud.

On désire un haut degré d'interactivité au niveau :

- de l'édition de la structure d'une scène,
- de la manipulation des caméras virtuelles,
- du positionnement ou de l'orientation des objets 3D,
- et de la définition et du positionnement des sources de lumière.

ED3D doit être un logiciel **évolutif** ; on désire ainsi que la structure de données d'une scène puisse servir de base à un **système d'animation**. Ce dernier permettra la réalisation de films d'animation réalistes (dessins animés, spots publicitaires, ...).

V.4.3 Les Concepts d'ED3D

A. Les Objets 3D

Les objets 3D sont modélisés à l'aide d'un modèle polyédrique basé sur des facettes triangulaires.

Le but du logiciel ED3D n'est pas la création interactive d'objets 3D, bien que cette fonction puisse être rajoutée par la suite. Les objets de la scène seront donc *importés* sous la forme de fichiers ASCII. Ces derniers sont créés au LIG soit par le logiciel de "Sculpting" SURFMAN (André LeBlanc - LIG de l'EPFL [LEBL 90]), soit par le système Mira (D. Thalmann - Université de Montréal [MAGN 83]). Ces fichiers ASCII comprennent une description tridimensionnelle d'un objet (comme par exemple un cylindre, une voiture, un visage,...) et fournissent par conséquent la liste des sommets, des arêtes, et des polygones de l'objet 3D, mais également une description du matériau à simuler pour celui-ci.

B. Les caméras virtuelles

Les caméras virtuelles permettent de visualiser, par projection parallèle ou perspective, toute ou une partie de la scène. Plusieurs caméras peuvent exister, et peuvent être visualisées (par un objet 3D représentant une caméra) dans la scène. On dispose donc de plusieurs points de vue de la scène simultanément. Une caméra virtuelle peut être schématisée comme, de nouveau, à la figure V.3.

C. Les sources de lumière

Les sources de lumière sont soit ponctuelles soit directionnelles. Leurs définitions permettent de créer un rendu réaliste sur la scène, à l'aide d'un modèle d'illumination à spécifier. Plusieurs sources lumineuses peuvent exister, et s'il existe des sources ponctuelles, elles peuvent être visualisées dans la scène par un objet 3D représentant une source de lumière.

V.4.4 La structure hiérarchique

A. Le concept de hiérarchie

Une scène est fréquemment composée de divers objets, parfois complexes. Les objets complexes sont alors formés de sous-objets plus simples, qui sont eux-mêmes décomposés jusqu'à atteindre les primitives graphiques de base ; les facettes, les arêtes, et les sommets. On façonne de la sorte une **structure hiérarchique**.

Une telle organisation est particulièrement intéressante lorsque des objets sont composés de parties semblables (comme précédemment pour l'exemple du vélo). Chaque sous-objet, faisant partie d'un objet complexe, est défini dans son propre repère. Sa position, son orientation et son facteur d'échelle (en réalité sa *matrice de modeling*) ne sont pas définis directement, mais par rapport au sous-objet situé juste au niveau supérieur dans la hiérarchie, ou plus précisément par rapport au repère propre à ce sous-objet. Ce dernier sera appelé le sous-objet *père*. Le passage d'un niveau au niveau suivant ou précédent de la hiérarchie s'obtient en effectuant un **changement de repère**. Ce changement peut s'exprimer par une matrice de transformation. Cette association d'un repère à chaque sous-objet permet de modéliser très facilement les objets articulés. En effet, puisque chaque sous-objet est lié à un sous-objet de niveau supérieur, celui-ci suivra automatiquement les déplacements de son sous-objet *père*. Le déplacement des objets et leur articulation sont réalisés en modifiant uniquement les matrices de changement de repère, c'est-à-dire en déplaçant les repères. (voir figure V.8)

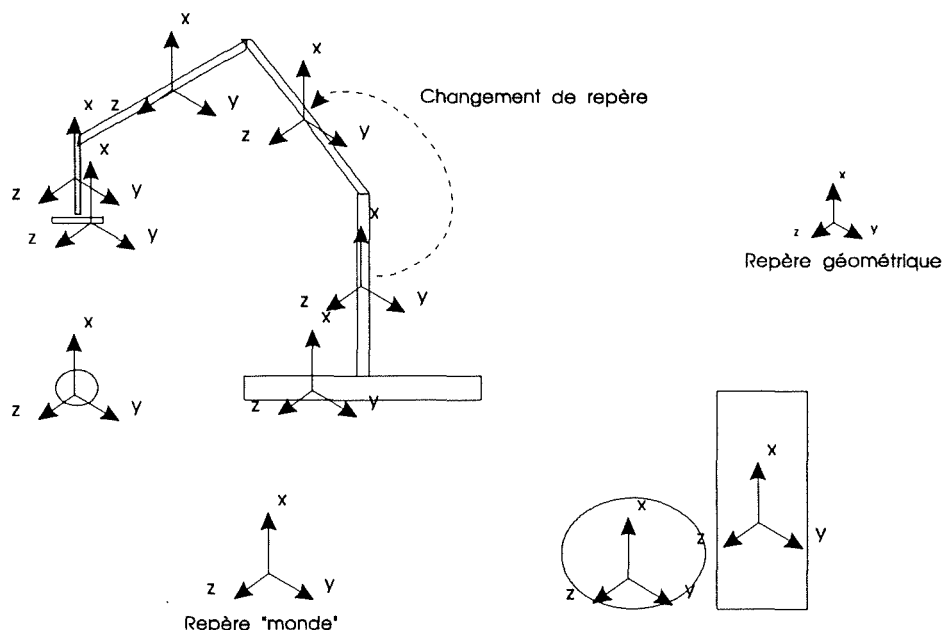


Figure V.8 : Les repères géométriques d'une scène 3D.

Il serait également intéressant de définir les caméras virtuelles et les sources de lumière de manière à ce qu'elles puissent faire partie intégrante de cette structure hiérarchique. On appellera dès lors les objets 3D, les sources de lumière et les caméras virtuelles les *éléments* d'une scène. Un *élément* est par conséquent un modèle dont l'intégration au sein de la hiérarchie est envisageable (et ceci sans a priori sur une méthodologie de programmation particulière). La position et l'orientation de ces éléments dans la scène doivent pouvoir être modifiées interactivement. Chaque élément est défini, comme expliqué précédemment, dans un repère propre. Par contre, la position, l'orientation de ce repère sont définis par une matrice de transformation et par rapport au repère de l'élément défini à un niveau supérieur.

Pour réaliser l'affichage correct de la scène, il suffit alors de parcourir l'arbre de données graphiques en tenant compte des matrices de changement de repère (voir de nouveau figure V.8), pour disposer des coordonnées de tous les objets de la scène dans un repère unique, appelé *repère monde*.

B. Le Design de la Hiérarchie arborescente

Une hiérarchie est destinée à intégrer les informations nécessaires au niveau d'une scène. Je citerai ici les critères principaux dans le design de cette hiérarchie arborescente, et qui soient utilisables ultérieurement pour un système d'animation [BOUL 90].

1) Les Critères retenus

Critère d'homogénéité : L'intégration d'objets variés et hétérogènes au sein de la hiérarchie doit se faire en découplant les informations de structure de la hiérarchie (les informations géométriques et topologiques de la scène) des informations externes (les informations fonctionnelles : type de l'élément, modèles de visualisation,...) qui lui sont rattachées. Ce principe de base a plusieurs avantages. Tout d'abord de rendre la structure de la hiérarchie - *l'armature* -, autonome des autres informations qui peuvent lui être rattachées. Dans une optique de division de travail, on peut envisager par exemple qu'un "designer" puisse ainsi travailler sur *l'habillage* d'une armature pourvue des surfaces par défaut d'un mannequin. De même une autre personne s'intéressera à la gestion du mouvement du mannequin en ne gardant que l'armature elle-même et en y ajoutant des éléments de perception, de contrainte, de trajectoire à suivre dans l'espace.

Critère de transparence : Le choix d'une représentation arborescente pour l'armature ne doit pas limiter la manipulation d'un objet complexe. Aussi, on doit pouvoir connaître facilement la position relative de deux éléments quelconques de la hiérarchie, et définir la situation absolue de l'objet complexe par l'intermédiaire de n'importe quel élément de l'armature.

2) La Mise en oeuvre

La brique de base pour la construction d'une hiérarchie arborescente est le *noeud*. Il rassemble les informations topologiques, géométriques et fonctionnelles d'un élément de la hiérarchie. Il permet de construire des **arborescences n-aires** (voir figure V.9) ; un noeud peut avoir autant de descendants (de noeuds *fils*) qu'il le désire.

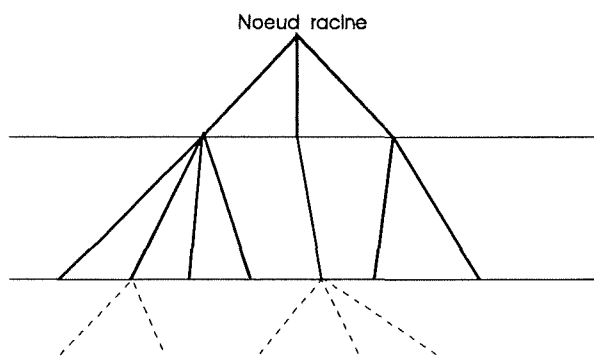


Figure V.9 : Une arborescence n-aire.

D'un point de vue géométrique, chaque noeud conserve l'information d'une transformation de *modeling* (appelée souvent *transformation locale*) vis-à-vis de son noeud *père*, qui est unique. Cette transformation définit le passage du repère courant vers son repère *père*.

Enfin, d'un point de vue fonctionnel, chaque noeud est pourvu d'une information de type qui lui permet d'associer des propriétés complémentaires et de répondre au critère d'homogénéité défini précédemment. On retrouve les types **Objet 3D**, **Caméra**, **Lumière**, et **Intermédiaire**, qui seront définis par la suite.

En maintenant à jour une matrice de transformation courante, une seule transformation (en fait, un changement de repère) par point sera requise. Cette matrice sera gérée à l'aide de la pile matricielle GL (explicitée au début de ce chapitre) de manière à faciliter le parcours de l'arbre. En se déplaçant en direction des *feuilles*, on sauvera à chaque noeud la matrice courante sur la pile, puis on la prémultipliera par la matrice correspondant au noeud. Lors du retour en arrière, il suffira de désempiler les matrices pour retrouver l'ancienne valeur (comme dans l'exemple du vélo).

Pour dessiner toute la structure de la scène, il faut donc, à chaque niveau, empiler la transformation dans la pile matricielle de GL, dessiner le noeud dans son repère propre, et enfin dépiler la transformation. Comme le parcours de l'arbre est réalisé le plus aisément par une procédure récursive, on obtiendra la gestion correcte de la matrice de transformation courante simplement en la déclarant comme un paramètre passé par valeur.

L'arbre de données graphiques se compose donc de noeuds typés. On peut noter également que la structure de ce type d'arbre (arité non fixée) peut être simulée à l'aide de listes.

3) L'intérêt

Les choix effectués pour la construction de la hiérarchie permettent d'obtenir très rapidement les transformations entre deux noeuds quelconques de la hiérarchie, et répondent donc au critère de transparence défini auparavant.

De plus, une telle conception est intéressante, notamment si l'on désire passer à la modélisation d'objets articulés et à l'animation de ceux-ci. Il faut cependant placer les repères nécessaires à l'animation attendue (par exemple, à la jointure de deux sous-objets comme à la figure V.8). Ces repères formeront le squelette de la structure à animer et sont à différencier des repères géométriques décrits ci-dessus, et qui ne correspondent pas forcément aux repères requis pour une animation judicieuse.

Il est intéressant de noter que, dans la pratique, on peut spécifier a priori l'armature de l'animation et ensuite *l'habiller*, ou à l'inverse, placer des objets dans une scène et définir a posteriori les repères pour l'animation.

V.4.5 Le modèle conceptuel des données

On définit ici les structures de données, les entités manipulées par le logiciel ED3D.

On distingue :

- la scène,
- la structure hiérarchique composée de différents types de noeuds,

On peut également citer les bibliothèques sur mémoire de masse :

- les bibliothèques des objets 3D,
- et les bibliothèques des scènes.

Examinons plus en détail ces différentes entités ...

A. La Scène

Une scène est définie par :

- un identifiant,
- une hiérarchie arborescente, à savoir : un noeud racine, et des noeuds de différents types (Objet, Caméra, Lumière, et Intermédiaire),
- la liste des noeuds Caméra, et
- la liste des noeuds Lumière définis dans la scène.

B. La hiérarchie arborescente

La structure arborescente est composée de noeuds typés. Ceux-ci conservent plusieurs informations :

- une matrice de transformation locale qui spécifie une position, une orientation et un facteur d'échelle de l'élément dans le repère de son noeud *père*,
- la référence de son noeud *père*,
- la liste de ses noeuds *fil*s.

Chacun de ces noeuds est accompagné d'informations complémentaires selon le type. On distingue :

-Le type Intermédiaire, qui ne contient aucune information supplémentaire, et constitue simplement un repère intermédiaire.

-Le type **Objet**, qui contient les informations définissant une surface polygonale dans son repère local, c'est-à-dire :

- un identifiant,
- une liste des facettes de l'objet ; une facette étant définie par une liste de trois sommets et un vecteur normal, un sommet étant défini par un vecteur 3D et un vecteur normal.
- un matériau prédéfini pour la surface de l'objet,
- un modèle d'illumination à appliquer à l'objet ; soit en Fil-de-Fer, soit en couleurs pleines, soit en illumination constante de Lambert, soit en illumination avec lissage de Gouraud.

-Le type **Camera**, qui définit une caméra virtuelle, et qui correspond intuitivement à l'oeil de l'observateur (voir de nouveau la figure V.3). Elle est définie par :

- un identifiant,
- un angle d'ouverture,
- les plans "*near*" et "*far*",
- un type de projection (parallèle ou perspective).

- Le type "**Lumière**", qui définit une source lumineuse par :

- un identifiant,
- un type (ponctuelle ou directionnelle),
- une couleur (une valeur RGB),
- une intensité,
- une position (si source ponctuelle) ou une direction (si source directionnelle),
- une figure 3D.

D'autres types de noeuds peuvent être développés en fonction des besoins (Contraintes, Solides, ...).

C. Les librairies des objets 3D

Ces librairies contiennent la description des objets 3D, et sont stockées sur mémoire de masse. Le couple (Nom de la librairie, Nom d'objet) doit être unique, afin d'identifier univoquement un objet dans une librairie. Comme je l'ai déjà précisé auparavant, la description des objets 3D d'une scène est considérée comme un fichier ASCII *Input* d'ED3D. A titre d'information, un tel fichier, créé par le logiciel SURFMAN [LEBL 90] et décrivant un objet 3D particulier (en l'occurrence, la "tête" d'Elvis !), est annexé à la fin de ce mémoire (Annexe 1).

D. Les bibliothèques de scènes

Les bibliothèques de scènes constituent l'ensemble des fichiers stockées sur mémoire de masse, et décrivant une scène 3D. Ces fichiers sont produits par ED3D. Leur format de stockage est ASCII, car leurs informations doivent pouvoir être lues facilement pour le "designer" de la scène. Celles-ci comprennent la description de toute la hiérarchie arborescente d'une scène. Pour information, un fichier décrivant une scène particulière est annexé à la fin de ce mémoire (Annexe 2).

V.4.6 Spécifications fonctionnelles

Avant de présenter les fonctionnalités offertes par ED3D, je donnerai tout d'abord un aperçu de l'interface utilisée.

A. Interface

Pour assurer une édition *interactive* de scènes, ED3D fournit une interface utilisateur agréable et naturelle. Le clavier n'est utile que lors de l'introduction de chaînes de caractères. La souris est le périphérique d'entrée privilégié du logiciel. Elle permet de sélectionner l'option désirée dans un menu, de désigner un objet dans la scène. Elle permet également de contrôler en temps réel les mouvements de caméras, et d'opérer interactivement des rotations et des translations sur un objet préalablement sélectionné. Une *boîte à boutons* permet de régler interactivement certains paramètres de contrôle de la caméra, tel le "Zoom".

L'édition d'une structure hiérarchique 3D peut s'avérer très difficile à partir de projections sur un écran 2D. Il est donc judicieux d'afficher constamment à l'usage de l'utilisateur une représentation 2D de la géométrie de la structure hiérarchique qu'il construit. Il pourra ainsi déceler directement l'élément courant de la hiérarchie, et éventuellement activer un élément, en sélectionnant directement sur la représentation 2D de son noeud. Il s'est avéré, en pratique, que la visualisation simultanée de la scène et de cet outil constitue une aide appréciable.

L'écran est l'unique périphérique de sortie et est composé de plusieurs fenêtres de visualisation, d'une fenêtre de description 2D de la hiérarchie, et d'un menu de fonctions. Les fenêtres de visualisation sont réservées aux opérations de visualisation de scènes. Elles sont le résultat d'une projection géométrique, et donc relatives à une position d'un observateur, d'une caméra virtuelle. Il y aura donc autant de fenêtres de visualisation que de caméras virtuelles définies dans la scène. Une *fenêtre de hiérarchie* est destinée à symboliser, par un arbre 2D, la structure de données hiérarchique de la scène. Le menu permet de choisir la fonction désirée et fait référence à la fenêtre de visualisation active.

B. Fonctionnalités

La fonction principale d'ED3D est de modifier une scène stockée sur la mémoire de masse ou en créer une nouvelle.

Dans le cas d'une modification, on spécifie le nom de la scène désirée, dans la liste fournie par le menu. Elle est directement visualisée à l'écran, et on retrouve les mêmes fonctionnalités décrites ci-dessous, dans le cas de la création d'une scène.

Les possibilités offertes sont :

- La création d'un noeud dans la hiérarchie.
- La sélection d'un noeud de la hiérarchie.
- Le mouvement d'un noeud sélectionné.
- Le déplacement de noeuds dans la hiérarchie.
- La duplication de noeuds dans la hiérarchie.
- L'effacement de noeuds de la hiérarchie.
- La visualisation de la scène.
- La sauvegarde de la scène.

Il est important de noter qu'il existe constamment un noeud privilégié, le *noeud courant*. C'est celui sur lequel les fonctions de *modification* vont s'effectuer. Passons à présent en revue ces différentes fonctions...

1) La création d'un noeud dans la hiérarchie

On peut créer plusieurs types de noeuds : un noeud *Objet*, un noeud *Lumière*, un noeud *Camera*, ou un noeud *Intermédiaire*. Il suffit de choisir l'option de création correspondante dans le menu.

Le noeud créé est alors inséré dans la structure de données hiérarchique, en tant que *fil* du noeud courant.

Dans le cas d'un nouveau noeud *Objet 3D*, la figure géométrique correspondante est directement visualisée dans la scène, après avoir été lue sur fichier.

Dans le cas d'un nouveau noeud *Lumière*, on choisit également son type (ponctuelle ou infinie) et sa couleur (choix dans une liste fournie par menu). La nouvelle source est directement positionnée et visualisée dans la scène (si elle est ponctuelle) à un endroit prédéfini. Son effet sur l'illumination de la scène est immédiat. Il est à remarquer qu'une faible lumière ambiante existe par défaut.

Dans le cas d'un nouveau noeud *Intermédiaire*, seul un repère 3D est visualisé dans la scène (par 3 axes x, y, z disposés de manière conventionnelle).

Dans le cas d'un nouveau noeud *Caméra*, celle-ci est visualisée directement par une figure représentant une caméra dans la scène. Sa position et ses attributs sont créés par défaut. La projection résultante est visualisée dans une nouvelle fenêtre, et la nouvelle caméra créée devient active.

Dans tous les cas, la structure de données hiérarchique s'est enrichie du noeud créé et est réaffichée dans la fenêtre de la représentation 2D de la hiérarchie.

2) La sélection d'un noeud de la hiérarchie

Il est possible de sélectionner un élément (*Objet 3D*, *Lumière*, *Caméra* ou *Intermédiaire*) avec la souris directement dans la scène, ou sur la représentation de l'élément dans la fenêtre de représentation 2D de la hiérarchie.

Dans le cas d'un élément *Caméra*, on peut simplement porter le curseur dans une fenêtre de visualisation, et la caméra relative à celle-ci devient automatiquement active.

3) Le mouvement d'un noeud sélectionné

Une fois sélectionné, un noeud peut subir des transformations géométriques.

Les rotations sont effectuées à l'aide de la souris selon le modèle de la sphère virtuelle, basé sur [CHEN 88]. L'utilisateur peut alors imaginer que l'objet qu'il a sélectionné est compris dans une sphère qu'il peut déplacer ; il semble alors faire "rouler" l'objet en appliquant des mouvements à la souris. Les translations peuvent être opérées horizontalement (mouvement horizontal de la souris), ou verticalement (mouvement vertical de la souris).

Ces translations et ces rotations sont effectuées par rapport au repère de la caméra active, afin d'avoir une perception naturelle de la transformation appliquée.

Dans le cas où l'on sélectionne un objet ayant des descendants, ces derniers subissent évidemment, par la loi de l'héritage, tous ces mouvements.

4) Le déplacement de noeuds dans la hiérarchie

On peut choisir un déplacement **relatif** ou **absolu** au sein de la structure hiérarchique.

Dans le cas d'un déplacement relatif, le repère *fils* sélectionné est redéfini par

rapport à un repère *père* qui est à spécifier, mais il garde son positionnement relatif antérieur (par rapport à son ancien *père*).

Dans le cas d'un déplacement absolu, le repère *fils* sélectionné est redéfini par rapport à un repère *père* qui est à spécifier, mais il garde son positionnement absolu (aucune modification de la position géométrique 3D du repère *fils*).

Ces déplacements sont effectués en redéfinissant la matrice de positionnement du repère¹².

5) La duplication d'un noeud sélectionné

Il est fréquent qu'un même élément soit utilisé plusieurs fois dans une scène, par exemple une fenêtre dans un immeuble (ou comme les roues dans l'exemple du vélo). Deux politiques de gestion de la répétition de ce type d'élément sont possibles : la **copie** ou l'**instanciation**.

La *copie* comme son nom l'indique conduit à la duplication de toute la structure de données représentant l'élément. Si cet élément doit être modifié par la suite, il faudra alors modifier l'une après l'autre chacune de ses copies. Cette solution n'est donc guère intéressante dans les applications basées sur la multiplication d'éléments de base identiques.

L'*instanciation* d'un élément consiste à ne dupliquer que la matrice de transformation correspondante, c'est-à-dire le noeud auquel est raccroché le sous-arbre dont l'élément est la racine. Ceci permet d'avoir des clones de tailles différentes, orientés et positionnés séparément. La description de l'élément reste unique, elle est simplement accédée (via plusieurs pointeurs) pour l'affichage. Outre l'avantage d'éviter un gaspillage de la mémoire, cette technique permet de modifier simultanément toutes les instances d'un élément.

Dans le cas où seule une instance de l'élément dupliqué doit être modifiée, celle-ci sera transformée en copie par duplication de sa description et pourra alors être modifiée séparément. Il est important de souligner que le passage inverse, de copie à instance, n'est en principe pas possible, vu qu'il est difficile de déterminer de quel modèle une copie est issue lorsqu'elle a été modifiée.

¹²Ces changements de positionnement des repères géométriques ont été explicités lors de l'étude de la structure hiérarchique en début de ce chapitre.

Si le noeud concerné est la racine d'un sous-arbre, tout ce dernier fait l'objet de la copie ou de l'instanciation. Une représentation schématique de ces fonctions est donnée à la figure V.10

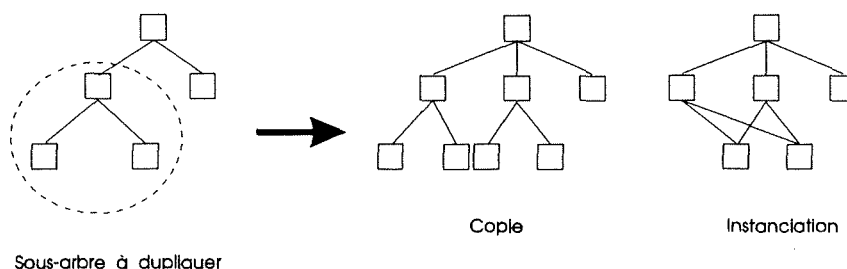


Figure V.10 : Deux politiques de duplication de noeuds : la copie et l'instanciation.

La fonction d'instanciation n'était pas prévue au début de la réalisation du logiciel ED3D, car elle nécessitait une modification importante de la structure de données hiérarchiques. En effet, cette duplication permettait alors qu'un noeud puisse posséder plusieurs noeuds "père" !! Une nouvelle structure a donc été spécifiée et mise en oeuvre pour tenir compte de cet effet.

6) L'effacement de noeuds de la hiérarchie

Il suffit, une fois un noeud sélectionné, de choisir la fonction d'effacement dans le menu. Si le noeud concerné est la racine d'un sous-arbre, tout ce dernier fait l'objet de l'effacement.

7) La visualisation de la scène

Le but est de contrôler le résultat de l'algorithme de visualisation.

On peut sélectionner différents modes de visualisation. Les modes de représentation possibles sont le modèle *fil-de-fer*, en couleurs pleines, en illumination constante de Lambert, et en illumination avec lissage de Gouraud. Par défaut, le mode est *fil-de-fer*, et est appliqué à toute la scène.

La caméra virtuelle active peut être contrôlée en modifiant son type de projection, un "Zoom",... Il est possible également d'en modifier la position et l'orientation de manière interactive.

Trois types de mouvement sont possibles :

1) On vise constamment un noeud, un objet ou par défaut la scène entière. On effectue des rotations de la caméra à l'aide de la souris, selon le modèle de la sphère virtuelle [CHEN 88] : l'utilisateur peut alors s'imaginer qu'il "tourne autour" du noeud visé, comme s'il se déplaçait sur une bulle. On peut se rapprocher ou s'éloigner du point d'intérêt (du noeud) grâce à un mouvement de souris.

2) On effectue des rotations dans le repère de la caméra. On peut déplacer la caméra horizontalement (mouvement horizontal de la souris), ou verticalement (mouvement vertical de la souris). On peut se rapprocher ou s'éloigner selon sa direction de vision par mouvements de souris.

3) On effectue des translations du repère de la caméra par rapport au repère de référence. On déplace la vue de la caméra en effectuant des translations horizontales ou verticales par mouvements de souris.

On peut noter que, pour faciliter le travail du *designer* de la scène, ED3D permet de visualiser sélectivement les différents repères des éléments composant la scène, sous la forme de 3 axes orthogonaux deux à deux X Y Z.

8) La sauvegarde de la scène

Il est évidemment possible de mémoriser une description complète de la scène créée ou modifiée dans la librairie des scènes.

V.4.7 Evolution du logiciel

Le but premier d'ED3D réside en l'édition interactive de scènes 3D réalistes. Les extensions possibles sont nombreuses. On peut envisager :

A. Au point de vue de la synthèse d'objets 3D

- La création d'objets modélisés par des facettes planes, par les méthodes d'extrusion et de révolution étudiées dans le chapitre III. On peut ensuite étendre la méthode de création en prévoyant la possibilité de modifier interactivement tout sommet d'un objet, éventuellement selon une fonction de déformation (on fait alors du "Sculpting").

- La Création d'objets paramétriques par surfaces de Bézier, *Splines*, ou encore en utilisant des surfaces fractales, ou des systèmes de particules pour la modélisation d'objets *flous*.

- La possibilité de "*Mapping*" de textures pour simuler la rugosité de certaines surfaces.

- La Création d'un *éditeur de matériau*. Le concepteur de scènes 3D aura, grâce à ce type d'outil, la possibilité de créer ses propres matériaux pour les objets 3D d'une scène, en spécifiant les paramètres nécessaires : Couleur de diffusion, couleur de spéculaire, largeur de spéculaire, degré de transparence,...

Toutes ces techniques ont été exposées et référencées dans le chapitre IV de ce mémoire.

B. Au point de vue de l'édition de scènes 3D

- Obtenir un contrôle précis des distances, et un placement programmé d'objets. Ceci faciliterait notamment l'élaboration d'une structure hiérarchique destinée à un système d'animation.

- Proposer d'autres types de lumières (tubes fluorescents, spots,...).

C. Au point de vue du rendu réaliste de scènes 3D

- Obtenir un rendu plus réaliste, en utilisant d'autres modèles d'illumination, ou un "*Ray-tracer*". Ce type d'outil permet, en fournissant toutes les informations concernant la scène et la position d'un observateur, d'y appliquer un algorithme de *lancer de rayons*. Il produit alors une image de la scène, correspondant à la vision de cet observateur, avec un rendu réaliste de haute qualité.

D. Au point de vue de l'animation de scènes 3D

- Il est possible, de par la conception de la structure hiérarchique d'ED3D, de créer une animation de la scène. Il suffit de définir des repères d'animation sur tout noeud de la hiérarchie, et ensuite de définir l'évolution (en fait des matrices de changement de repère) en fonction du temps, c'est-à-dire définir le *script* de la scène. Cette méthode est connue sous le nom de *système d'animation algorithmique* [THAL 88].

- Créer un film d'animation avec un rendu réaliste de qualité. Il suffit pour cela de produire les séquences (les images) de l'animation de la scène en utilisant le "*Ray-tracer*". Ces images peuvent être alors enregistrées, par exemple tous les 25^{èmes} de seconde, sur bande magnétique vidéo pour générer le film d'animation.

V.4.8 Commentaires sur l'implémentation

Le logiciel ED3D a été développé en adoptant une approche Top-Down. Une découpe modulaire des fonctions présentées ci-dessus a permis une implémentation incrémentale, ponctuée par de nombreux tests.

La réalisation d'ED3D n'a pas été de tout repos... En effet, excepté l'apprentissage accéléré des "finesses" du langage C employé pour la programmation de ce logiciel, certains concepts importants de la bibliothèque GL étaient au début, pour un novice en infographie, difficiles à maîtriser. Par exemple, le concept de *Caméra virtuelle*, des différents systèmes de coordonnées, et bien d'autres encore ...

V.5 Conclusion

L'objectif de l'implémentation d'un logiciel dédié à la synthèse de scènes 3D réalistes, tel qu'ED3D, a été pleinement réalisé. Cela m'a permis en effet de mettre en pratique les concepts théoriques étudiés lors du stage, et de me rendre compte des difficultés inhérentes à ce type d'application. C'est également grâce à l'aide précieuse des membres du LIG dans ce domaine, qu'ED3D a été mené à bien dans les détails prévus. Les quelques photographies commentées à la fin de ce chapitre et une vidéo VHS couleur de 20 minutes peuvent donner un aperçu des fonctionnalités du logiciel réalisé.

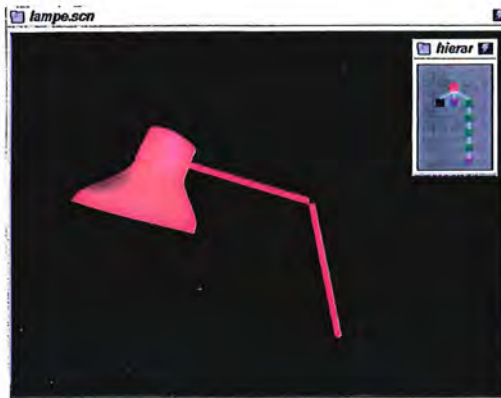
Les photos ci-dessous tentent de donner un bref aperçu des fonctions du logiciel d'édition hiérarchique de scènes 3D réalistes "ED3D". Réalisé sur station Silicon Graphics POWER Iris 4D-GT par A. Arcuri, LIG-EPFL, Lausanne 1990.



La visualisation d'un objet 3D peut s'effectuer selon divers degrés de réalisme ; le modèle fil-de-fer, le modèle d'illumination ambiante, le modèle d'illumination de Lambert, le modèle d'illumination avec lissage de Gouraud. On peut remarquer également les différents repères géométriques 3D de la scène.



Une lampe, par exemple, se prête bien à une modélisation hiérarchique. Les sous-parties inférieures seront définies relativement au repère de leur sous-partie supérieure. La hiérarchie 2D illustre cette structure arborescente.



La visualisation des repères géométriques de tous les éléments de la scène peut être utile lorsque l'on applique des transformations géométriques (rotations ou translations) sur un ou plusieurs éléments de la scène, ou lors de la modification de la structure arborescente. La fenêtre dans le coin supérieur droit symbolise, en 2D, la structure hiérarchique de la scène. Grâce à la définition d'une nomenclature de couleurs, on peut distinguer directement les différents types de noeuds et leur position relative dans la hiérarchie.



VI . CONCLUSION

Après un bref historique de l'infographie dans le chapitre II, nous avons abordé dans le chapitre suivant les principales méthodes de modélisation d'objets dans l'espace, le moyen de les disposer géométriquement dans une scène tridimensionnelle, et de définir un champ de vision en spécifiant la position d'un observateur particulier.

Nous avons étudié ensuite, dans le chapitre IV, les techniques qui permettent de créer une image d'une troublante ressemblance avec la réalité, comme le prouvent les illustrations en couleurs présentes dans ce mémoire. Nous avons vu comment synthétiser des sources lumineuses et leurs effets sur les objets de la scène. Nous avons également traité des effets spéciaux que l'on peut ajouter à une scène : comment simuler des ombres, des objets transparents, ou des textures de surfaces irrégulières sur les objets qui composent la scène.

En chaque point de cette scène, les différentes contributions des sources lumineuses et d'un éventuel éclairage ambiant seront prises en compte pour déterminer quelle quantité de lumière est émise, transmise ou réfléchi en direction de l'oeil de l'observateur. Selon le degré de réalisme que l'on se propose d'atteindre ou le degré de complexité que l'on s'interdit de dépasser, on pourra se limiter à la simple réflexion diffuse, y ajouter des termes correcteurs tenant compte de la nature de la surface réfléchissante (réflexion spéculaire), ou alors se poser des questions sur l'histoire du rayon incident (de combien de réflexions, et sur quels objets, est-il l'aboutissement ?), comme l'algorithme du lancer de rayons ou celui du calcul de radiosit . Bien entendu, cela se paie en temps calcul tr s long, pour ne pas dire interminables...

A ce sujet, il faut pr ciser que les temps calcul d'une image varient en fonction de la complexit  de la sc ne et de la qualit  recherch e ; cela va de quelques minutes   plusieurs jours... Certaines images de synth se pour la t l vision ont parfois occup  un supercalculateur   temps plein pendant des heures. Nous nous trouvons l  devant l'un des principes de base de l'image de synth se, applicable quelle que soit la puissance de l'ordinateur : Plus l'exigence en r alisme est grande, plus le temps de calcul est  lev . Le choix d'une technique de rendu r aliste est d termin  avant tout sur ce principe.

Dans le dernier chapitre, la r alisation d'un  diteur hi rarchique de sc nes 3D a  t  pr sent e. Ce projet a  t  men  dans le cadre du stage de fin d' tudes au laboratoire d'infographie de l'Ecole Polytechnique F d rale de Lausanne. Celui-ci m'a permis de me rendre compte des probl mes pratiques des techniques infographiques expos es dans ce m moire.

Les techniques de synthèse d'images 3D réalistes permettent de visualiser des scènes à un instant donné. Or, l'intérêt de beaucoup de scènes réelles réside dans leur évolution au cours du temps. C'est ainsi que les systèmes d'animation, qui consistent en la modification d'une scène au cours du temps, prennent de plus en plus d'importance. Ainsi, l'outil implémenté personnellement au LIG de Lausanne est destiné à évoluer vers un système d'animation... Si les infographistes se sentent de plus attirés vers les systèmes d'animation 3D, ils se heurtent cependant à cette limite décidément peu compressible : le temps. Aussi pour l'instant, l'animation image par image traditionnelle reste d'application. Mais grâce aux recherches effectuées dans le domaine du parallélisme et les architectures multi-processeurs, on peut espérer un jour avoir le moyen de créer une animation 3D réaliste en temps réel.

"From bar charts to wire-frame models to photo-realism, when you're speaking of the graphics world, one line says it all : You've come a long way, baby !"

J.M. Tazelaar, in "Computer Graphics : State of the Art", Byte December 1990.

VII . BIBLIOGRAPHIE

[AKEL 89] K. AKELEY, "The Silicon Graphics 4D/GTX SuperWorkstation", IEEE Computer Graphics and Applications, 1989.

[ANSI 85b] ANSI, American National Standard for Information Processing Systems - Computer Graphics - Graphical Kernel System (GKS) Functional Description, ANSI X3.124-1985, ANSI, New York, 1985.

[ANSI 88] ANSI, American National Standard for Information Processing Systems - Computer Graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) Functional Description, Archive File Format, Clear-Text Encoding of Archive File, ANSI X3.144-1988, ANSI, New York, 1988.

[APGA 88] B. APGAR, B. BERSACK, A. MAMMEN, "A display system for the Stellar Graphics SuperComputer Model gs1000", Proceedings of SIGGRAPH 88, 1988.

[APPE 68] A. APPEL, "Some techniques for shading machine rendering of solids", Proceedings" SJCC, 1968.

[ATHE 78] P. ATHERTON, K. WEILER, D. GREENBERG, "Polygon Shadow Generation", Proceedings of SIGGRAPH 78, 1978.

[BARS 83] B. BARSKY, J. BEATTY, "Local Control of Bias and Tension in Beta-Splines", ACM Transactions on Graphics, 2 (2), April 1983.

[BEZI 87] P. BEZIER, "Courbes et Surfaces", Hermès, Série Mathématique et CAO, vol. 4, 1987.

[BLIN 76] J.F. BLINN, M.E. NEWELL, "Texture and reflection in computer generated images", Communications of ACM, 19, 10, 1976.

[BLIN 77] J.F. BLINN, "Models of light reflection for computer synthesized pictures", ACM computer Graphics, Vol 11, No 2, 1977.

[BLIN 82] J.F. BLINN, "Light reflection functions for simulations of clouds and dusty surfaces", In ACM Computer Graphics, 16, 3, July 1982.

[BORD 89] B. S. BORDEN, "Graphics Processing on a Graphics SuperComputer", Computer Graphics and Applications, 1989.

[BOUL 90] R. BOULIC, "Mise en oeuvre d'une hiérarchie pour l'animation", Note technique, EPFL, 1990.

[BRET 88] M. BRET, "Images de Synthèse", Ed. Bo-Pré, Dunod Informatique, 1988.

[BROS 85] A. BROSSARD, R. CAMARERO, D. THALMANN, "Espace tridimensionnels de textures", Ecole Polytechnique de Montréal, EPM/RT-85-27, 1985.

- [BROT 84] L. BROTMAN, N.I. BADLER, "Generating soft shadows with a depth buffer algorithm", IEEE Computer Graphics and Applications, vol. 4, n° 10, 1984.
- [BUSE 88] J.P. BUSER, "Modélisation, transformations géométriques 2D et 3D", Note du cours postgrade en informatique technique - Visualisation scientifique et simulation graphique EPFL, Lausanne 1988, et dans [LIEB 88], chapitre 3, p. 123.
- [CATM 74] E. CATMULL, "A subdivision algorithm for computer display of curved surfaces", Ph.D. Dissertation, University of Utah, 1974.
- [CHEN 88] M. CHEN, S.J. MOUNTFORD, A. SELLEN, "A study in interactive 3D rotation using 2D control devices", ACM Computer Graphics, 22, 4, August 1988.
- [COHE 80] M.F. COHEN, T. LYCHE, R.F. RIESENFELD, "Discrete B-Splines and subdivision techniques in computer-aided geometric design and computer graphics", Computer Graphics and Image Processing, 14, 2, 1980.
- [COHE 85] M.F. COHEN, D.P. GREENBERG, "The hemi-cube. A radiosity solution for complex environment", Computer Graphics, 19, 3, 1985.
- [COOK 81] R.L. COOK, "A reflectance model for realistic image synthesis", Master's Thesis, Cornell University, Ithaca, 1981.
- [COOK 82] R.L. COOK, K.E. TORRANCE, "A reflectance model for computer graphics", 15, 3, 1981. In ACM transactions on Graphics, 1, 1, 1982.
- [CROW 77] F.C. CROW, "Shadow Algorithms for Computer Graphics", vol. 11, n°2, 1977.
- [FOLE 90] J.D. FOLEY, A. van DAM, S.K. FEINER, J.F. Hughes, "Computer Graphics - Principles and Practice", Second Edition, Ed. Addison Wesley, 1990.
- [GOUR 71] H. GOURAUD, "Computer display of curved surfaces", University of Utah Comp. Sc. Dept., Utec-Csc, June 1971.
- [GSPC 79] Graphics Standards Planning Committee, Status Report of the GSPC, Computer Graphics, 13 (3), August 1979.
- [HALL 83] R.A. HALL, D.P. GREENBERG, "A testbed for realistic image synthesis", IEEE Computer Graphics and Applications, vol. 3, n° 8, 1983.
- [INTE 88] International Standards Organization, International Standard Information Processing Systems - Computer Graphics - Graphical Kernel System for Three Dimensions (GKS - 3D), Functional Description, ISO Document Number 8805:1988(E), American Standards Institute, New York, 1988.
- [KAJI 84] J.T. KAJIYA, B.P. Von HERZEN, "Ray tracing volume densities", Proceedings of SIGGRAPH'84, Computer Graphics, vol. 18, n° 3, 1984.

- [KAY 79] D.S. KAY, "Transparency, Refraction and Ray Tracing for Computer Synthesized Images", M.S. Thesis, Program of Computer Graphics, Cornell University, Ithaca, New York, January 1979.
- [KAWA 82] Y. KAWAGUCHI, "A morphological study of the form of nature", In ACM Computer Graphics, 16, 223-232, July 1982.
- [LEBL 90] A. LEBLANC, "Surfman : User's reference manual", Note Technique LIG, EPFL, Lausanne, 1990.
- [LIEB 88] T.M. LIEBLING, "Infographie et Applications", Ed. Masson, 1988.
- [MAGN 83] N. MAGNENAT-THALMANN, D. THALMANN, "The use of high level graphical types in the MIRA animation system", IEEE Computer Graphics and Applications, 3, 9, 1983.
- [MAGN 87] N. MAGNENAT-THALMANN, D. THALMANN, "The direction of synthetic actors in the film Rendez-vous à Montréal", IEEE Computer Graphics and Applications, 7, 12, 1987.
- [MAND 75] B. MANDELBROT, "Les objets fractals", Nouvelle Bibliothèque Scientifique, Flammarion, 1975.
- [MEGA 89] Megatek Corporation, "Sigma 70 - Advanced Graphics Workstations", Megatek Corporation, San Diego, 1989.
- [NEWE 72] M.E. NEWELL, R.G. NEWELL, T.L. SANCHI, "A solution to the hidden-surface problem", Proceedings of ACM National Conference, 1972.
- [NEWM 79] W.M. NEWMANN, R.F. SPROULL, "Principles of Interactive Computer Graphics", Ed. Mc Graw Hill, N.Y., 1973, 2e Ed., 1979.
- [NISH 74] T. NISHITA, E. NAKAMAE, "An algorithm for half-toned representation of three-dimensional objects", Information Processing Society of Japan, vol. 14, 1974.
- [OPPE 86] P.E. OPPENHEIMER, "Real time design and animation of fractals plants and trees", Computer Graphics, 20, 4, 55-64, August 1986.
- [PERL 85] R. PERLIN, "An image synthesizer", Proceedings of SIGGRAPH'85, Computer Graphics, 1985.
- [PEAC 85] D.R. PEACHEY, "Solid Texturing of complex surfaces", Proceedings of SIGGRAPH 85, Computer Graphics, vol. 19, n° 3, 1985.
- [PEAC 86] D.R. PEACHEY, "Modeling waves and surf", Computer Graphics, 20, 4, 65-74, August 1986.
- [PHON 73] B.T. PHONG, "Illumination for computer-generated images", University of Utah, Computer Science Department, UTEC-CSc, 1973.

VII . Bibliographie

[PHON 75] B.T. PHONG, "Illumination for computer-generated pictures", Communications of ACM, 18, 6, 1975.

[REEV 83] W. REEVES, "Particle systems : a technique for modeling a class of fuzzy objects", Conception des systèmes de synthèse d'images, INRIA, 1983 ; ACM Transactions on Graphics, 2, 2, 1983.

[ROBE 64] L.G. ROBERTS, "Machine perception of three-dimensional solids", MIT Press, 1964.

[ROGE 85] D.F. ROGERS, "Procedural elements for computer graphics", Ed. McGraw-Hill, 1985.

[SCHU 69] R. SCHUMAKER, B. BRAND, M. GILLI, W. SHARP, "Study for applying computer generated images to visual simulation", General Electric Technical Information Series, 1969.

[SCHW 87] PH. SCHWEIZER, "Infographie I-II", Presses Polytechniques Romandes, 1987.

[SGUS 88] Silicon Graphics, "GT Graphics Library User's Guide", 1988.

[SUTH 74] I.E. SUTHERLAND, R.F. SPROULL, R.A. SCHUMACKER, "A characterization of ten-hidden-surface algorithm", Computer Survey, vol. 6, n° 1, 1974.

[THAL 88] D. THALMANN, "Synthèse et animation des images réalistes", Note du cours postgrade en informatique technique - Visualisation scientifique et simulation graphique EPFL, Lausanne 1988, et dans [LIEB 88], chapitre 8, p. 295.

[TORR 67] K.E. TORRANCE, E.M. SPARROW, "Theory for Off-Specular Reflection from Roughened Surfaces", J. Opt. Soc. Am., vol. 57, n° 9, 1967.

[WARN 68] J.E. WARNOCK, "A hidden line algorithm for half-tone picture representation", Technical Report Computer Science Department, University of Utah, 1968.

[WARN 69] J.E. WARNOCK, "A hidden-surface algorithm for computer generated half-tone pictures", Computer Science Department, University of Utah, 1969.

[WEIL 86] J. WEIL, "The synthesis of cloth objects", Computer Graphics, 20, 4, 49-54, August 1986.

[WHIT 80] J.T. WHITTED , "An improved illumination model for shaded display", Communications of ACM, vol.23, n°6, 1980.

[WILL 78] L. WILLIAMS, "Casting curved shadows on curved surfaces", Proceedings of SIGGRAPH 78, 1978.

ANNEXE 1

Cette annexe a pour but de donner une idée de ce que peut représenter la description géométrique tridimensionnelle d'un objet . Elle présente le contenu d'un fichier ASCII produit par le logiciel SURFMAN [LEBL 90]. Il comprend la liste des sommets (cfr. "Vertices"), des arêtes (cfr. "Edges"), et des facettes triangulaires (cfr. "Triangles") décrivant un objet 3D (en l'occurrence, la "tête" d'Elvis !), mais également la description du matériau à simuler et une matrice de positionnement initial de l'objet. Ce fichier est considéré comme un *Input* pour le logiciel d'édition de scènes "ED3D".

SURFACE

[VERTICES 317

{(0,[10.902599,17.839951,7.560309])}(0,[11.527678,5.877051,11.614880])}(0,[8.736016,0.136875,11.121791])}(0,[11.330900,-6.251218,10.470602])}(0,[9.402793,10.465504,10.985413])}(0,[8.815630,-14.712977,10.349617])}(0,[7.328949,-18.132206,10.596279])}(0,[13.893839,26.580507,8.604719])}(0,[12.935011,15.572908,4.761972])}(0,[13.907121,10.061535,6.341402])}(0,[16.012318,-1.027064,1.829171])}(0,[15.595020,-5.134429,-0.952908])}(0,[13.974039,-15.056467,-1.399113])}(0,[12.543054,-17.819771,-0.217932])}(0,[11.747459,-31.842276,-6.760890])}(0,[18.100494,28.983215,-5.386805])}(0,[20.518152,18.388433,0.223347])}(0,[17.070148,-1.084236,-3.990459])}(0,[14.867474,-10.377068,-12.924517])}(0,[13.815455,-16.026974,-18.101315])}(0,[13.164454,-32.573219,-11.316436])}(0,[9.809628,32.232437,2.817747])}(0,[9.490624,33.739830,-15.051995])}(0,[17.198099,26.342203,-25.058075])}(0,[16.208721,16.148771,-30.294758])}(0,[15.151465,5.691792,-30.547323])}(0,[13.672261,-4.792322,-29.723866])}(0,[13.098492,-11.042540,-27.089638])}(0,[9.462169,-16.161549,-26.170671])}(0,[11.706313,-31.222832,-25.905060])}(0,[0.000000,24.687555,16.061810])}(0,[0.000000,31.930603,12.404165])}(0,[0.000000,16.931078,14.242912])}(0,[0.000000,6.349378,16.662462])}(0,[0.000000,3.769796,15.924469])}(0,[0.000000,-7.542233,19.699150])}(0,[0.000000,-8.222123,17.326708])}(0,[0.000000,-11.474643,17.184772])}(0,[0.000000,-16.680994,15.261778])}(0,[0.000000,-20.850952,15.835025])}(0,[0.000000,-24.974882,1.149481])}(0,[0.000000,36.299271,-7.109389])}(0,[0.000000,33.090233,-27.252602])}(0,[0.000000,23.193523,-34.755817])}(0,[0.000000,13.556787,-36.579681])}(0,[0.000000,3.449934,-36.191048])}(0,[0.000000,-6.075361,-34.256546])}(0,[0.000000,-15.604990,-31.054754])}(0,[0.000000,-16.763773,-29.001429])}(0,[4.015383,1.842675,11.581605])}(0,[3.565871,-7.886147,14.034953])}(0,[5.500789,-13.579891,11.908771])}(0,[0.000000,-15.876814,16.273832])}(0,[14.929338,-6.733703,-6.336489])}(0,[0.000000,-31.924049,-4.420652])}(0,[0.000000,-32.993053,-28.127020])}(0,[14.016123,-16.244116,-10.078685])}(0,[0.000000,-23.461876,14.622158])}(0,[10.137918,-21.505550,-1.225376])}(0,[6.868690,-23.136442,0.602604])}(0,[12.424315,-19.386183,-4.321825])}(0,[13.190269,-32.940857,-17.886896])}(0,[13.482546,0.925997,5.789557])}(0,[8.896776,1.189774,11.770349])}(0,[19.554054,1.422950,-12.216813])}(0,[15.509310,6.057036,0.351066])}(0,[3.367502,-23.620356,6.576515])}(0,[6.627646,-11.251101,11.887404])}(0,[10.097178,-20.947468,5.351504])}(0,[14.631285,-2.757112,4.624376])}(0,[12.353975,-14.312981,5.304744])}(0,[13.808616,-8.249975,4.542978])}(0,[8.712763,2.726206,12.035456])}(0,[21.428389,7.736059,-9.224857])}(0,[19.850405,2.821634,-22.493383])}(0,[0.000000,-12.497030,16.514683])}(0,[0.000000,-13.395276,16.176676])}(0,[1.856552,-11.145938,16.433615])}(0,[9.695533,-20.760914,0.634920])}(0,[2.495107,-15.565265,15.334345])}(0,[1.699029,-12.218231,15.481383])}(0,[10.598444,1.883770,9.607301])}(0,[6.482950,2.906870,11.945389])}(0,[5.125891,-22.617447,2.282424])}(0,[16.681541,-0.649482,-0.531110])}(0,[16.917156,-5.088922,-12.880918])}(0,[16.510994,2.731690,-0.654137])}(0,[0.000000,-6.365374,20.257278])}(0,[2.197641,-16.068731,13.638480])}(0,[4.548233,-18.791609,12.594794])}(0,[21.267588,25.006533,-14.073202])}(0,[3.128366,-12.775035,14.081857])}(0,[8.567169,5.756806,14.340894])}(0,[6.751270,18.040384,12.241084])}(0,[12.164427,-18.057442,-2.073051])}(0,[13.863974,-15.123775,-6.588678])}(0,[13.753861,-15.177435,-19.605610])}(0,[6.324451,-15.469187,-28.808126])}(0,[16.423523,-9.408113,-14.546780])}(0,[21.084845,13.712193,-23.325262])}(0,[6.400972,1.208052,12.063647])}(0,[18.254938,2.454523,-7.363019])}(0,[2.935128,-4.776243,14.730924])}(0,[1.236877,-7.421282,18.951143])}(0,[4.517402,-7.277534,13.603612])}(0,[4.211052,-3.825348,12.984076])}(0,[6.689789,-5.608907,12.535165])}(0,[17.018642,-3.071333,-10.236419])}(0,[3.800124,-12.580106,13.879871])}(0,[3.272498,4.230219,15.538642])}(0,[7.733701,4.418436,14.406209])}(0,[5.581463,5.546195,16.179787])}(0,[11.266379,4.248612,11.734109])}(0,[13.212272,4.032273,8.461245])}(0,[0.000000,-22.962399,8.982288])}(0,[0.000000,-24.556688,1.590637])}(0,[1.058087,-8.001549,16.440332])}(0,[3.301733,-7.362214,15.701899])}(0,[2.146042,-7.224884,17.414326])}(0,[5.932350,0.127800,12.074086])}(0,[13.939073,-14.941992,-4.268023])}(0,[18.103943,1.428318,-5.200167])}(0,[3.912262,-5.345820,13.813013])}(0,[0.000000,13.580142,15.045197])}(0,[6.203393,13.130712,14.040094])}(0,[11.614095,13.483271,9.40

1141)) (0,[0.000000,-8.992071,16.739351]) (0,[0.000000,-7.166533,20.184805])
 (0,[1.164665,-8.816724,16.427778]) (0,[9.743611,2.292403,11.159297]) (0,[9.788703,
 1.453389,10.889975]) (0,[4.777865,2.110051,11.467116]) (0,[4.810446,1.638532,11.48453
 6]) (0,[2.725379,-13.575791,14.062132]) (0,[1.620441,-14.754728,16.025070])
 (0,[0.000000,-14.633720,16.686419]) (0,[2.306065,3.109835,15.449661]) (0,[1.073738,-
 2.534648,17.381426]) (0,[1.641110,-5.222827,17.753983]) (0,[1.262013,1.610553,
 15.627025]) (0,[0.000000,-0.727488,17.769711]) (0,[0.000000,-4.130205,19.091393])
 (0,[2.133555,-6.715880,18.431318]) (0,[3.491355,-6.715035,16.211321]) (0,[4.129887,-
 6.806156,14.795761]) (0,[0.000000,-10.479492,16.953932]) (0,[1.401314,-10.130985,
 16.240978]) (0,[3.516374,-10.587273,13.832363]) (0,[4.823107,-13.652906,12.419521])
 (0,[3.502354,-14.177380,14.429721]) (0,[4.673386,-14.429981,13.193293]) (0,[6.524371,-
 8.328068,12.604507]) (0,[4.314329,-23.568150,11.641802]) (0,[7.148369,-22.991491,
 9.046897]) (0,[2.974884,-2.329006,13.562474]) (0,[4.669536,-6.126135,13.009899])
 (0,[11.943521,-2.925533,9.456804]) (0,[10.877830,0.460226,9.262537]) (0,[0.000000,
 18.834251,15.723783]) (0,[7.452693,19.624672,12.553356]) (0,[12.779366,18.365120,7.55
 7693]) (0,[15.208416,16.045830,4.398127]) (0,[15.413902,10.450072,5.001947]) (0,[17.529
 066,6.119012,-0.264795]) (0,[18.063747,3.267267,-2.121682]) (0,[20.331944,3.802442,-
 7.923785]) (0,[20.621805,1.893129,-13.755104]) (0,[1.055401,-5.953936,19.646204])
 (0,[1.290872,-6.944960,19.545219]) (0,[5.017402,-10.481950,12.741875]) (0,[11.214912,-
 10.385231,8.584795]) (0,[13.280635,-5.403389,7.603832]) (0,[0.000014,-22.594646,
 15.496573]) (0,[2.284484,-22.289738,13.614668]) (0,[6.514051,-20.948042,11.124460])
 (0,[9.356365,-19.575232,7.874773]) (0,[16.738924,-1.734040,-6.895063]) (0,[10.902599,
 17.839951,7.560309]) (0,[11.527678,5.877051,11.614880]) (0,[8.736016,0.136875,
 11.121791]) (0,[11.330900,-6.251218,10.470602]) (0,[9.402793,-10.465504,10.985413])
 (0,[8.815630,-14.712977,10.349617]) (0,[7.328949,-18.132206,10.596279]) (0,[13.893839,
 26.580507,8.604719]) (0,[12.935011,15.572908,4.761972]) (0,[13.907121,
 10.061535,6.341402]) (0,[16.012318,-1.027064,1.829171]) (0,[15.595020,-5.134429,-
 0.952908]) (0,[13.974039,-15.056467,-1.399113]) (0,[12.543054,-17.819771,-
 0.217932]) (0,[11.747459,-31.842276,-6.760890]) (0,[18.100494,28.983215,-
 5.386805]) (0,[20.518152,18.388433,0.223347]) (0,[17.070148,-1.084236,-3.990459]) (0,[14.867474,
 -10.377068,-12.924517]) (0,[13.815455,-16.026974,-18.101315]) (0,[13.164454,-32.573219,
 -11.316436]) (0,[9.809628,32.232437,2.817747]) (0,[9.490624,33.739830,-15.051995])
 (0,[17.198099,26.342203,-25.058075]) (0,[16.208721,16.148771,-30.294758]) (0,[15.151465,
 5.691792,-30.547323]) (0,[13.672261,-4.792322,-29.723866]) (0,[13.098492,-11.042540,
 -27.089638]) (0,[9.462169,-16.161549,-26.170671]) (0,[11.706313,-31.222832,-25.905060])
 (0,[4.015383,1.842675,11.581605]) (0,[3.565871,-7.886147,14.034953]) (0,[5.500789,-
 13.579891,11.908771]) (0,[14.929338,-6.733703,-6.336489]) (0,[14.016123,-16.244116,
 -10.078685]) (0,[10.137918,-21.505550,-1.225376]) (0,[6.868690,-23.136442,0.602604])
 (0,[12.424315,-19.386183,-4.321825]) (0,[13.190269,-32.940857,-17.886896]) (0,[13.482546,
 0.925997,5.789557]) (0,[8.896776,1.189774,11.770349]) (0,[19.554054,1.422950,-
 12.216813]) (0,[15.509310,6.057036,0.351066]) (0,[3.367502,-23.620356,6.576515])
 (0,[6.627646,-11.251101,11.887404]) (0,[10.097178,-20.947468,5.351504]) (0,[14.631285,
 -2.757112,4.624376]) (0,[12.353975,-14.312981,5.304744]) (0,[13.808616,-8.249975,
 4.542978]) (0,[8.712763,2.726206,12.035456]) (0,[21.428389,7.736059,-9.224857])
 (0,[19.850405,2.821634,-22.493383]) (0,[1.856552,-11.145938,16.433615]) (0,[9.695533,-
 20.760914,0.634920]) (0,[2.495107,-15.565265,15.334345]) (0,[1.699029,-12.218231,
 15.481383]) (0,[10.598444,1.883770,9.607301]) (0,[6.482950,2.906870,11.945389])
 (0,[5.125891,-22.617447,2.282424]) (0,[16.681541,-0.649482,-0.531110]) (0,[16.917156,-
 5.088922,-12.880918]) (0,[16.510994,2.731690,-0.654137]) (0,[2.197641,-16.068731,
 13.638480]) (0,[4.548233,-18.791609,12.594794]) (0,[21.267588,25.006533,-14.073202])
 (0,[3.128366,-12.775035,14.081857]) (0,[8.567169,5.756806,14.340894]) (0,[6.751270,
 18.040384,12.241084]) (0,[12.164427,-18.057442,-2.073051]) (0,[13.863974,-15.123775,-
 6.588678]) (0,[13.753861,-15.177435,-19.605610]) (0,[6.324451,-15.469187,-
 28.808126]) (0,[16.423523,-9.408113,-14.546780]) (0,[21.084845,13.712193,-
 23.325262]) (0,[6.400972,1.208052,12.063647]) (0,[18.254938,2.454523,-
 7.363019]) (0,[2.935128,-4.776243,14.730924]) (0,[1.236877,-7.421282,
 18.951143]) (0,[4.517402,-7.277534,13.603612]) (0,[4.211052,-3.825348,
 12.984076]) (0,[6.689789,-5.608907,12.535165]) (0,[17.018642,-3.071333,-10.236419])

(0,[-3.800124,-12.580106,13.879871])(0,[-3.272498,4.230219,15.538642])(0,[-7.733701,4.418436,14.406209])(0,[-5.581463,5.546195,16.179787])(0,[-11.266379,4.248612,11.734109])(0,[-13.212272,4.032273,8.461245])(0,[-1.058087,-8.001549,16.440332])(0,[-3.301733,-7.362214,15.701899])(0,[-2.146042,-7.224884,17.414326])(0,[-5.932350,0.127800,12.074086])(0,[-13.939073,-14.941992,-4.268023])(0,[-18.103943,1.428318,-5.200167])(0,[-3.912262,-5.345820,13.813013])(0,[-6.203393,13.130712,14.040094])(0,[-11.614095,13.483271,9.401141])(0,[-1.164665,-8.816724,16.427778])(0,[-9.743611,2.292403,11.159297])(0,[-9.788703,1.453389,10.889975])(0,[-4.777865,2.110051,11.467116])(0,[-4.810446,1.638532,11.484536])(0,[-2.725379,-13.575791,14.062132])(0,[-1.620441,-14.754728,16.025070])(0,[-2.306065,3.109835,15.449661])(0,[-1.073738,-2.534648,17.381426])(0,[-1.641110,-5.222827,17.753983])(0,[-1.262013,1.610553,15.627025])(0,[-2.133555,-6.715880,18.431318])(0,[-3.491355,-6.715035,16.211321])(0,[-4.129887,-6.806156,14.795761])(0,[-1.401314,-10.130985,16.240978])(0,[-3.516374,-10.587273,13.832363])(0,[-4.823107,-13.652906,12.419521])(0,[-3.502354,-14.177380,14.429721])(0,[-4.673386,-14.429981,13.193293])(0,[-6.524371,-8.328068,12.604507])(0,[-4.314329,-23.568150,11.641802])(0,[-7.148369,-22.991491,9.046897])(0,[-2.974884,-2.329006,13.562474])(0,[-4.669536,-6.126135,13.009899])(0,[-11.943521,-2.925533,9.456804])(0,[-10.877830,0.460226,9.262537])(0,[-7.452693,19.624672,12.553356])(0,[-12.779366,18.365120,7.557693])(0,[-15.208416,16.045830,4.398127])(0,[-15.413902,10.450072,5.001947])(0,[-17.529066,6.119012,-0.264795])(0,[-18.068747,3.267267,-2.121682])(0,[-20.331944,3.802442,-7.923785])(0,[-20.621805,1.893129,-13.755104])(0,[-1.055401,-5.953936,19.646204])(0,[-1.290872,-6.944960,19.545219])(0,[-5.017402,-10.481950,12.741875])(0,[-11.214912,-10.385231,8.584795])(0,[-13.280635,-5.403389,7.603832])(0,[-2.284484,-22.289738,13.614668])(0,[-6.514051,-20.948042,11.124460])(0,[-9.356365,-19.575232,7.874773])(0,[-16.738924,-1.734040,-6.895063])

EDGES 913

{(31,30,0)(30,7,0)(7,31,0)(159,0,0)(0,160,0)(160,159,0)(124,111,0)(111,92,0)(92,124,0)(151,104,0)(104,67,0)(67,151,0)(106,151,0)(151,4,0)(4,106,0)(145,37,0)(37,77,0)(77,145,0)(77,75,0)(75,80,0)(80,77,0)(172,57,0)(57,173,0)(173,172,0)(0,161,0)(161,160,0)(0,125,0)(125,8,0)(8,0,0)(62,171,0)(171,69,0)(69,62,0)(5,70,0)(70,170,0)(170,5,0)(70,175,0)(175,68,0)(68,70,0)(83,40,0)(40,59,0)(59,83,0)(7,21,0)(21,31,0)(15,7,0)(7,16,0)(16,15,0)(161,8,0)(8,9,0)(9,161,0)(162,9,0)(9,65,0)(65,162,0)(11,12,0)(12,120,0)(120,11,0)(13,94,0)(94,120,0)(120,13,0)(94,95,0)(95,120,0)(16,90,0)(90,15,0)(163,162,0)(65,163,0)(24,99,0)(99,74,0)(74,24,0)(98,18,0)(18,96,0)(96,98,0)(27,98,0)(96,27,0)(19,20,0)(20,61,0)(61,19,0)(41,31,0)(21,41,0)(21,22,0)(22,41,0)(23,42,0)(42,22,0)(22,23,0)(23,24,0)(24,43,0)(43,23,0)(24,25,0)(25,45,0)(45,24,0)(25,26,0)(26,45,0)(26,97,0)(97,46,0)(46,26,0)(28,55,0)(55,48,0)(48,28,0)(158,32,0)(32,93,0)(93,158,0)(32,123,0)(123,93,0)(49,131,0)(131,109,0)(109,49,0)(103,35,0)(35,36,0)(36,103,0)(67,169,0)(169,51,0)(51,67,0)(38,89,0)(89,88,0)(88,38,0)(152,174,0)(174,173,0)(173,152,0)(125,9,0)(65,113,0)(113,62,0)(62,65,0)(69,10,0)(10,62,0)(11,71,0)(71,12,0)(70,13,0)(13,12,0)(12,70,0)(15,21,0)(164,86,0)(86,84,0)(84,164,0)(17,11,0)(11,53,0)(53,17,0)(18,53,0)(53,56,0)(56,18,0)(56,19,0)(19,18,0)(15,22,0)(15,23,0)(90,23,0)(25,74,0)(74,26,0)(26,98,0)(27,26,0)(19,29,0)(29,28,0)(28,19,0)(42,41,0)(43,42,0)(24,44,0)(44,43,0)(45,44,0)(46,45,0)(97,47,0)(47,46,0)(48,97,0)(97,28,0)(33,34,0)(34,109,0)(109,33,0)(144,50,0)(50,104,0)(104,144,0)(116,126,0)(126,128,0)(128,116,0)(51,5,0)(5,67,0)(51,6,0)(6,5,0)(119,105,0)(105,2,0)(2,119,0)(79,52,0)(52,38,0)(38,79,0)(85,53,0)(18,85,0)(53,120,0)(95,53,0)(40,54,0)(54,59,0)(29,55,0)(56,60,0)(60,20,0)(20,56,0)(66,114,0)(114,115,0)(115,66,0)(94,58,0)(58,60,0)(60,94,0)(68,83,0)(83,78,0)(78,68,0)(58,59,0)(59,14,0)(14,58,0)(54,14,0)(14,60,0)(14,20,0)(61,29,0)(63,119,0)(2,63,0)(81,157,0)(157,62,0)(62,81,0)(96,28,0)(97,96,0)(97,27,0)(86,163,0)(65,86,0)(10,65,0)(153,66,0)(66,68,0)(68,153,0)(152,57,0)(57,66,0)(66,152,0)(67,4,0)(5,4,0)(78,59,0)(58,78,0)(152,153,0)(153,174,0)(94,78,0)(157,156,0)(156,62,0)(171,71,0)(71,69,0)(170,71,0)(171,170,0)(11,69,0)(11,10,0)(70,71,0)(68,13,0)(6,70,0)(110,82,0)(82,72,0)(72,110,0)(113,81,0)(64,165,0)(165,101,0)(101,64,0)(99,73,0)(73,74,0)(85,166,0)(166,64,0)(64,85,0)(166,165,0)(76,135,0)(135,134,0)(134,76,0)(147,77,0)(77,108,0)(108,147,0)(37,75,0)(78,13,0)(148,149,0)(149,150,0)(150,148,0)(6,150,0)(150,88,0)(88,6,0)(80,91,0)(91,77,0)(92,110,0)(110,10,0)(1,92,0)(130,63,0)(2,130,0)(112,110,0)(72,112,0)(66,83,0)(115,83,0)(84,11,0)(17,84,0)(84,10,0)(10,86,0)(74,98,0)(121,101,0)(101,164,0)(164,121,0)(35,168,0)(168,127,0)(127,

35,0)(88,79,0)(51,150,0)(89,6,0)(38,39,0)(39,89,0)(16,73,0)(73,90,0)(99,90,0)(91,108,0)(12
 4,125,0)(125,93,0)(93,124,0)(0,93,0)(159,158,0)(93,159,0)(110,109,0)(109,82,0)(60,95,0)(5
 6,95,0)(19,96,0)(48,47,0)(85,98,0)(98,166,0)(23,99,0)(105,122,0)(122,155,0)(155,105,0)(17
 ,164,0)(17,176,0)(176,121,0)(121,17,0)(102,137,0)(137,138,0)(138,102,0)(139,154,0)(154,
 49,0)(49,139,0)(117,116,0)(116,50,0)(50,117,0)(144,117,0)(36,116,0)(116,103,0)(154,105,
 0)(105,49,0)(119,49,0)(105,106,0)(106,2,0)(155,106,0)(106,156,0)(156,2,0)(3,106,0)(4,3,0)(
 107,85,0)(64,107,0)(107,53,0)(91,51,0)(51,108,0)(51,147,0)(110,111,0)(111,109,0)(109,13
 6,0)(136,49,0)(111,33,0)(112,129,0)(129,81,0)(81,112,0)(112,1,0)(123,124,0)(112,113,0)(1
 13,1,0)(9,113,0)(9,1,0)(57,114,0)(115,40,0)(118,103,0)(116,118,0)(128,50,0)(142,118,0)(11
 8,143,0)(143,142,0)(118,117,0)(117,143,0)(100,119,0)(63,100,0)(132,119,0)(100,132,0)(17
 6,101,0)(105,102,0)(102,122,0)(144,143,0)(123,33,0)(33,124,0)(92,125,0)(1,125,0)(36,126,
 0)(168,87,0)(87,127,0)(146,145,0)(77,146,0)(147,146,0)(72,129,0)(2,157,0)(157,130,0)(49,
 132,0)(132,131,0)(131,82,0)(133,76,0)(134,133,0)(135,52,0)(52,134,0)(79,134,0)(51,148,0)
 (34,136,0)(136,139,0)(34,139,0)(140,137,0)(137,139,0)(139,140,0)(137,141,0)(141,138,0)(
 142,103,0)(34,140,0)(137,154,0)(140,141,0)(141,167,0)(167,138,0)(142,138,0)(167,142,0)(
 143,102,0)(102,142,0)(143,122,0)(144,122,0)(144,155,0)(142,168,0)(168,103,0)(126,145,0)
 (146,126,0)(146,128,0)(146,50,0)(147,50,0)(169,147,0)(148,133,0)(133,149,0)(134,149,0)(
 79,149,0)(79,150,0)(169,50,0)(151,155,0)(155,104,0)(102,154,0)(156,171,0)(3,156,0)(81,1
 30,0)(159,30,0)(30,158,0)(159,7,0)(160,7,0)(160,16,0)(161,16,0)(162,161,0)(162,16,0)(163,
 16,0)(163,73,0)(164,163,0)(74,166,0)(73,166,0)(73,165,0)(165,164,0)(73,164,0)(87,167,0)(
 141,87,0)(167,168,0)(169,104,0)(4,170,0)(170,3,0)(3,171,0)(39,172,0)(173,39,0)(173,89,0)(
 174,89,0)(174,6,0)(153,175,0)(175,174,0)(6,175,0)(53,176,0)(184,30,0)(31,184,0)(301,177,
 0)(177,300,0)(300,301,0)(243,262,0)(262,272,0)(272,243,0)(221,255,0)(255,293,0)(293,221
 ,0)(181,293,0)(293,257,0)(257,181,0)(229,37,0)(145,229,0)(232,75,0)(75,229,0)(229,232,0)
 (313,57,0)(172,313,0)(302,177,0)(301,302,0)(185,273,0)(273,177,0)(177,185,0)(223,312,0)
 (312,216,0)(216,223,0)(311,224,0)(224,182,0)(182,311,0)(222,315,0)(315,224,0)(224,222,0
)(213,40,0)(40,235,0)(235,213,0)(198,184,0)(31,198,0)(193,184,0)(184,192,0)(192,193,0)(1
 86,185,0)(185,302,0)(302,186,0)(219,186,0)(186,303,0)(303,219,0)(269,189,0)(189,188,0)(
 188,269,0)(269,245,0)(245,190,0)(190,269,0)(246,245,0)(269,246,0)(241,193,0)(192,241,0)
 (303,304,0)(304,219,0)(228,250,0)(250,201,0)(201,228,0)(247,195,0)(195,249,0)(249,247,0
)(249,204,0)(204,247,0)(215,197,0)(197,196,0)(196,215,0)(41,198,0)(41,199,0)(199,198,0)(
 199,42,0)(42,200,0)(200,199,0)(43,201,0)(201,200,0)(200,43,0)(45,202,0)(202,201,0)(201,
 45,0)(45,203,0)(203,202,0)(46,248,0)(248,203,0)(203,46,0)(55,205,0)(205,48,0)(244,32,0)(
 158,244,0)(244,123,0)(260,277,0)(277,207,0)(207,260,0)(35,254,0)(254,36,0)(209,310,0)(3
 10,221,0)(221,209,0)(239,240,0)(240,38,0)(38,239,0)(313,314,0)(314,294,0)(294,313,0)(18
 6,273,0)(216,264,0)(264,219,0)(219,216,0)(187,223,0)(216,187,0)(189,225,0)(225,188,0)(1
 89,190,0)(190,224,0)(224,189,0)(198,192,0)(236,238,0)(238,305,0)(305,236,0)(210,188,0)(
 188,194,0)(194,210,0)(211,210,0)(210,195,0)(195,211,0)(195,196,0)(196,211,0)(199,192,0)
 (200,192,0)(200,241,0)(203,228,0)(228,202,0)(249,203,0)(203,204,0)(205,206,0)(206,196,0
)(196,205,0)(44,201,0)(47,248,0)(205,248,0)(248,48,0)(260,34,0)(33,260,0)(255,208,0)(208
 ,287,0)(287,255,0)(274,126,0)(126,265,0)(265,274,0)(182,209,0)(221,182,0)(182,183,0)(18
 3,209,0)(179,256,0)(256,268,0)(268,179,0)(52,231,0)(231,38,0)(210,237,0)(237,195,0)(269
 ,210,0)(210,246,0)(213,54,0)(55,206,0)(197,214,0)(214,211,0)(211,197,0)(114,220,0)(220,
 115,0)(214,212,0)(212,245,0)(245,214,0)(230,235,0)(235,222,0)(222,230,0)(191,213,0)(213
 ,212,0)(212,191,0)(191,54,0)(214,191,0)(197,191,0)(206,215,0)(268,217,0)(217,179,0)(216
 ,299,0)(299,233,0)(233,216,0)(205,247,0)(247,248,0)(204,248,0)(304,238,0)(238,219,0)(21
 9,187,0)(222,220,0)(220,295,0)(295,222,0)(220,57,0)(57,294,0)(294,220,0)(181,221,0)(181
 ,182,0)(213,230,0)(230,212,0)(295,294,0)(314,295,0)(230,245,0)(216,298,0)(298,299,0)(22
 5,312,0)(223,225,0)(225,311,0)(311,312,0)(223,188,0)(187,188,0)(225,224,0)(190,222,0)(2
 24,183,0)(226,234,0)(234,261,0)(261,226,0)(233,264,0)(252,306,0)(306,218,0)(218,252,0)(
 228,227,0)(227,250,0)(218,307,0)(307,237,0)(237,218,0)(306,307,0)(280,135,0)(76,280,0)(
 259,229,0)(229,289,0)(289,259,0)(190,230,0)(292,291,0)(291,290,0)(290,292,0)(239,292,0)
 (292,183,0)(183,239,0)(242,232,0)(229,242,0)(178,261,0)(261,243,0)(243,178,0)(217,276,0
)(276,179,0)(261,263,0)(263,226,0)(235,220,0)(235,115,0)(188,236,0)(236,194,0)(187,236,
 0)(238,187,0)(249,228,0)(305,252,0)(252,270,0)(270,305,0)(127,309,0)(309,35,0)(231,239,
 0)(292,209,0)(183,240,0)(240,39,0)(227,193,0)(241,227,0)(241,250,0)(259,242,0)(244,273,
 0)(273,272,0)(272,244,0)(244,177,0)(158,300,0)(300,244,0)(260,261,0)(234,260,0)(246,214
 ,0)(246,211,0)(247,196,0)(307,249,0)(249,237,0)(250,200,0)(297,271,0)(271,256,0)(256,29

7,0)(305,194,0)(270,316,0)(316,194,0)(194,270,0)(283,282,0)(282,253,0)(253,283,0)(207,296,0)(296,284,0)(284,207,0)(208,265,0)(265,266,0)(266,208,0)(266,287,0)(265,36,0)(254,265,0)(207,256,0)(256,296,0)(207,268,0)(179,257,0)(257,256,0)(257,297,0)(179,298,0)(298,257,0)(257,180,0)(180,181,0)(237,258,0)(258,218,0)(210,258,0)(209,242,0)(259,209,0)(289,209,0)(260,262,0)(262,261,0)(207,281,0)(281,260,0)(33,262,0)(233,275,0)(275,263,0)(263,233,0)(178,263,0)(272,123,0)(178,264,0)(264,263,0)(264,186,0)(178,186,0)(254,267,0)(267,265,0)(208,274,0)(286,267,0)(267,285,0)(285,286,0)(266,267,0)(286,266,0)(268,251,0)(251,217,0)(268,278,0)(278,251,0)(252,316,0)(271,253,0)(253,256,0)(286,287,0)(272,33,0)(273,243,0)(273,178,0)(87,309,0)(145,288,0)(288,229,0)(288,289,0)(275,226,0)(299,179,0)(276,299,0)(278,207,0)(277,278,0)(234,277,0)(76,279,0)(279,280,0)(280,52,0)(280,231,0)(290,209,0)(281,34,0)(284,281,0)(284,34,0)(284,282,0)(282,140,0)(140,284,0)(283,141,0)(141,282,0)(254,285,0)(296,282,0)(283,308,0)(308,141,0)(283,285,0)(285,308,0)(253,286,0)(285,253,0)(271,286,0)(271,287,0)(297,287,0)(254,309,0)(309,285,0)(126,288,0)(274,288,0)(208,288,0)(208,289,0)(289,310,0)(291,279,0)(279,290,0)(291,280,0)(291,231,0)(292,231,0)(208,310,0)(255,297,0)(297,293,0)(296,253,0)(312,298,0)(298,180,0)(276,233,0)(30,300,0)(184,300,0)(184,301,0)(193,301,0)(193,302,0)(302,303,0)(193,303,0)(193,304,0)(227,304,0)(304,305,0)(307,228,0)(307,227,0)(306,227,0)(305,306,0)(305,227,0)(308,87,0)(309,308,0)(255,310,0)(311,181,0)(180,311,0)(312,180,0)(39,313,0)(240,313,0)(240,314,0)(183,314,0)(315,295,0)(314,315,0)(315,183,0)(316,210,0)

TRIANGLES 592

{(31,30,7,0,"c0",0,1,2)(159,0,160,0,"c0",3,4,5)(124,111,92,0,"c1",6,7,8)(151,104,67,0,"c1",9,10,11)(106,151,4,0,"c1",12,13,14)(145,37,77,0,"c1",15,16,17)(77,75,80,0,"c2",18,19,20)(172,57,173,0,"c1",21,22,23)(160,0,161,0,"c0",4,24,25)(0,125,8,0,"c1",26,27,28)(62,171,69,0,"c1",29,30,31)(5,70,170,0,"c1",32,33,34)(70,175,68,0,"c1",35,36,37)(83,40,59,0,"c1",38,39,40)(31,7,21,0,"c0",2,41,42)(15,7,16,0,"c0",43,44,45)(161,8,9,0,"c0",46,47,48)(162,9,65,0,"c0",49,50,51)(11,12,120,0,"c1",52,53,54)(13,94,120,0,"c1",55,56,57)(120,94,95,0,"c1",56,58,59)(15,16,90,0,"c0",45,60,61)(163,162,65,0,"c0",62,51,63)(24,99,74,0,"c0",64,65,66)(98,18,96,0,"c0",67,68,69)(27,98,96,0,"c0",70,69,71)(19,20,61,0,"c1",72,73,74)(41,31,21,0,"c0",75,42,76)(21,22,41,0,"c0",77,78,76)(23,42,22,0,"c0",79,80,81)(23,24,43,0,"c0",82,83,84)(24,25,45,0,"c0",85,86,87)(25,26,45,0,"c0",88,89,86)(26,97,46,0,"c0",90,91,92)(28,55,48,0,"c1",93,94,95)(158,32,93,0,"c0",96,97,98)(32,123,93,0,"c1",99,100,97)(49,131,109,0,"c1",101,102,103)(103,35,36,0,"c1",104,105,106)(67,169,51,0,"c1",107,108,109)(38,89,88,0,"c1",110,111,112)(152,174,173,0,"c1",113,114,115)(125,9,8,0,"c1",116,47,27)(65,113,62,0,"c1",117,118,119)(62,69,10,0,"c1",31,120,121)(11,71,12,0,"c1",122,123,52)(70,13,12,0,"c1",124,125,126)(21,7,15,0,"c0",41,43,127)(161,0,8,0,"c0",24,28,46)(164,86,84,0,"c0",128,129,130)(17,11,53,0,"c1",131,132,133)(18,53,56,0,"c1",134,135,136)(56,19,18,0,"c1",137,138,136)(15,22,21,0,"c0",139,77,127)(15,23,22,0,"c0",140,81,139)(23,15,90,0,"c0",140,61,141)(25,74,26,0,"c0",142,143,88)(26,98,27,0,"c0",144,70,145)(19,29,28,0,"c1",146,147,148)(22,42,41,0,"c0",80,149,78)(42,23,43,0,"c0",79,84,150)(24,44,43,0,"c0",151,152,83)(24,45,44,0,"c0",87,153,151)(26,46,45,0,"c0",92,154,89)(46,97,47,0,"c0",91,155,156)(48,97,28,0,"c0",157,158,95)(33,34,109,0,"c1",159,160,161)(144,50,104,0,"c1",162,163,164)(116,126,128,0,"c1",165,166,167)(67,51,5,0,"c1",109,168,169)(51,6,5,0,"c1",170,171,168)(119,105,2,0,"c1",172,173,174)(79,52,38,0,"c1",175,176,177)(85,53,18,0,"c1",178,134,179)(53,120,95,0,"c1",180,59,181)(59,40,54,0,"c1",39,182,183)(28,29,55,0,"c1",147,184,93)(56,60,20,0,"c1",185,186,187)(19,56,20,0,"c1",137,187,72)(66,114,115,0,"c1",188,189,190)(94,58,60,0,"c1",191,192,193)(68,83,78,0,"c1",194,195,196)(58,59,14,0,"c1",197,198,199)(14,59,54,0,"c1",198,183,200)(58,14,60,0,"c1",199,201,192)(60,14,20,0,"c1",201,202,186)(19,61,29,0,"c1",74,203,146)(63,119,2,0,"c1",204,174,205)(81,157,62,0,"c1",206,207,208)(74,25,24,0,"c0",142,85,66)(96,28,97,0,"c0",209,158,210)(97,27,96,0,"c0",211,71,210)(86,163,65,0,"c0",212,63,213)(62,10,65,0,"c1",121,214,119)(153,66,68,0,"c1",215,216,217)(152,57,66,0,"c1",218,219,220)(151,67,4,0,"c1",11,221,13)(4,67,5,0,"c1",221,169,222)(78,59,58,0,"c1",223,197,224)(174,152,153,0,"c1",113,225,226)(78,58,94,0,"c1",224,191,227)(157,156,62,0,"c1",228,229,207)(69,171,71,0,"c1",30,230,231)(170,71,171,0,"c1",232,230,233)(71,11,69,0,"c1",122,234,231)(10,69,11,0,"c1",120,234,235)(71,170,70,0,"c1",232,33,236)(71,70,12,0,"c1",236,126,123)(70,68,13,0,"c1",37,237,124)(5,6,70,0,"c1",171,238,32)(110,82,72,0,"c1",239,240,241)(113,81,62,0,"c1",242,208,118)(

64,165,101,0,"c0",243,244,245)(99,73,74,0,"c0",246,247,65)(85,166,64,0,"c0",248,249,250)(64,166,165,0,"c0",249,251,243)(76,135,134,0,"c2",252,253,254)(147,77,108,0,"c1",255,256,257)(37,75,77,0,"c2",258,18,16)(13,68,78,0,"c1",237,196,259)(78,83,59,0,"c1",195,40,223)(148,149,150,0,"c2",260,261,262)(6,150,88,0,"c1",263,264,265)(77,80,91,0,"c2",20,266,267)(92,110,1,0,"c0",268,269,270)(130,63,2,0,"c1",271,205,272)(112,110,72,0,"c1",273,241,274)(68,66,83,0,"c1",216,275,194)(83,66,115,0,"c1",275,190,276)(84,11,17,0,"c1",277,131,278)(84,10,11,0,"c1",279,235,277)(65,10,86,0,"c1",214,280,213)(74,98,26,0,"c0",281,144,143)(86,10,84,0,"c1",280,279,129)(121,101,164,0,"c0",282,283,284)(35,168,127,0,"c1",285,286,287)(79,38,88,0,"c1",177,112,288)(51,150,6,0,"c1",289,263,170)(88,89,6,0,"c1",111,290,265)(89,38,39,0,"c1",110,291,292)(90,16,73,0,"c0",60,293,294)(90,73,99,0,"c0",294,246,295)(77,91,108,0,"c2",267,296,256)(124,125,93,0,"c1",297,298,299)(93,125,0,0,"c1",298,26,300)(159,158,93,0,"c0",301,98,302)(82,110,109,0,"c1",239,303,304)(13,78,94,0,"c1",259,227,55)(95,94,60,0,"c1",58,193,305)(95,60,56,0,"c1",305,185,306)(53,95,56,0,"c1",181,306,135)(18,19,96,0,"c0",138,307,68)(96,19,28,0,"c0",307,148,209)(27,97,26,0,"c0",211,90,145)(47,97,48,0,"c0",155,157,308)(85,98,166,0,"c0",309,310,248)(98,85,18,0,"c0",309,179,67)(23,99,24,0,"c0",311,64,82)(23,90,99,0,"c0",141,295,311)(105,122,155,0,"c1",312,313,314)(17,164,84,0,"c0",315,130,278)(17,176,121,0,"c1",316,317,318)(102,137,138,0,"c1",319,320,321)(139,154,49,0,"c1",322,323,324)(117,116,50,0,"c1",325,326,327)(144,117,50,0,"c1",328,327,162)(103,36,116,0,"c1",106,329,330)(154,105,49,0,"c1",331,332,323)(119,49,105,0,"c1",333,332,172)(105,106,2,0,"c1",334,335,173)(105,155,106,0,"c1",314,336,334)(106,156,2,0,"c1",337,338,335)(3,106,4,0,"c1",339,14,340)(107,85,64,0,"c1",341,250,342)(107,53,85,0,"c1",343,178,341)(108,91,51,0,"c2",296,344,345)(51,147,108,0,"c1",346,257,345)(110,111,109,0,"c0",347,348,303)(109,136,49,0,"c1",349,350,103)(111,110,92,0,"c0",347,268,7)(109,111,33,0,"c1",348,351,161)(112,129,81,0,"c1",352,353,354)(110,112,1,0,"c0",273,355,269)(93,123,124,0,"c1",100,356,299)(112,113,1,0,"c0",357,358,355)(112,81,113,0,"c1",354,242,357)(9,113,65,0,"c1",359,117,50)(9,1,113,0,"c1",360,358,359)(66,57,114,0,"c1",219,361,188)(83,115,40,0,"c1",276,362,38)(118,103,116,0,"c1",363,330,364)(116,128,50,0,"c1",167,365,326)(142,118,143,0,"c1",366,367,368)(143,118,117,0,"c1",367,369,370)(117,118,116,0,"c1",369,364,325)(100,119,63,0,"c1",371,204,372)(132,119,100,0,"c1",373,371,374)(120,53,11,0,"c1",180,132,54)(12,13,120,0,"c1",125,57,53)(176,101,121,0,"c1",375,282,317)(121,164,17,0,"c0",284,315,318)(105,102,122,0,"c1",376,377,312)(143,117,144,0,"c1",370,328,378)(123,33,124,0,"c1",379,380,356)(124,33,111,0,"c1",380,351,6)(124,92,125,0,"c1",8,381,297)(92,1,125,0,"c1",270,382,381)(125,1,9,0,"c1",382,360,116)(36,126,116,0,"c1",383,165,329)(127,168,87,0,"c1",286,384,385)(146,145,77,0,"c1",386,17,387)(146,77,147,0,"c1",387,255,388)(112,72,129,0,"c1",274,389,352)(130,2,157,0,"c1",272,390,391)(131,49,132,0,"c4",101,392,393)(131,82,109,0,"c1",394,304,102)(132,49,119,0,"c1",392,333,373)(133,76,134,0,"c2",395,254,396)(134,135,52,0,"c2",253,397,398)(52,79,134,0,"c2",175,399,398)(148,150,51,0,"c1",262,289,400)(109,34,136,0,"c1",160,401,349)(49,136,139,0,"c1",350,402,324)(34,139,136,0,"c1",403,402,401)(140,137,139,0,"c1",404,405,406)(137,141,138,0,"c1",407,408,320)(142,103,118,0,"c1",409,363,366)(139,34,140,0,"c1",403,410,406)(139,137,154,0,"c1",405,411,322)(140,141,137,0,"c1",412,407,404)(141,167,138,0,"c1",413,414,408)(142,138,167,0,"c1",415,414,416)(142,143,102,0,"c1",368,417,418)(102,138,142,0,"c1",321,415,418)(143,122,102,0,"c1",419,377,417)(143,144,122,0,"c1",378,420,419)(122,144,155,0,"c1",420,421,313)(142,168,103,0,"c1",422,423,409)(126,145,146,0,"c1",424,386,425)(146,128,126,0,"c1",426,166,425)(128,146,50,0,"c1",426,427,365)(50,146,147,0,"c1",427,388,428)(169,147,51,0,"c1",429,346,108)(148,133,149,0,"c2",430,431,260)(149,133,134,0,"c2",431,396,432)(134,79,149,0,"c2",399,433,432)(149,79,150,0,"c2",433,434,261)(150,79,88,0,"c1",434,288,264)(169,50,147,0,"c1",435,428,429)(151,155,104,0,"c1",436,437,9)(173,57,152,0,"c1",22,218,115)(153,152,66,0,"c1",225,220,215)(154,137,102,0,"c1",411,319,438)(102,105,154,0,"c1",376,331,438)(155,144,104,0,"c1",421,164,437)(106,155,151,0,"c1",336,436,12)(171,62,156,0,"c1",29,229,439)(156,106,3,0,"c1",337,339,440)(81,130,157,0,"c1",441,391,206)(157,2,156,0,"c1",390,338,228)(159,93,0,0,"c0",302,300,3)(159,30,158,0,"c0",442,443,301)(159,7,30,0,"c0",444,1,442)(159,160,7,0,"c0",5,445,444)(7,160,16,0,"c0",445,446,44)(16,160,161,0,"c0",446,25,447)(161,9,162,0,"c0",48,49,448)(161,162,16,0,"c0",448,449,447)(16,162,163,0,"c0",449,62,450)(163,73,16,0,"c0",451,293,450)(164,163,86,0,"c0",452,212,128)(98,74,166,0,"c0",281,453,310)(166,74,73,0,"c0",453,247,454)(73,165,166,0,"c0",455,251,454)(165,164,101,0,"c0",456,283,244)(165,73,164,0,"

c0",455,457,456)(164,73,163,0,"c0",457,451,452)(168,35,103,0,"c1",285,104,423)(87,167,141,0,"c1",458,413,459)(167,168,142,0,"c1",460,422,416)(167,87,168,0,"c1",458,384,460)(50,169,104,0,"c1",435,461,163)(169,67,104,0,"c1",107,10,461)(3,4,170,0,"c1",340,462,463)(170,4,5,0,"c1",462,222,34)(156,3,171,0,"c1",440,464,439)(3,170,171,0,"c1",463,23,3,464)(39,172,173,0,"c1",465,23,466)(173,89,39,0,"c1",467,292,466)(173,174,89,0,"c1",1,14,468,467)(89,174,6,0,"c1",468,469,290)(174,153,175,0,"c1",226,470,471)(175,153,68,0,"c1",470,217,36)(175,70,6,0,"c1",35,238,472)(6,174,175,0,"c1",469,471,472)(17,53,176,0,"c1",133,473,316)(184,30,31,0,"c0",474,0,475)(301,177,300,0,"c0",476,477,478)(243,262,272,0,"c1",479,480,481)(221,255,293,0,"c1",482,483,484)(181,293,257,0,"c1",485,486,487)(229,37,145,0,"c1",488,15,489)(232,75,229,0,"c2",490,491,492)(313,57,172,0,"c1",493,21,494)(302,177,301,0,"c0",495,476,496)(185,273,177,0,"c1",497,498,499)(223,312,216,0,"c1",500,501,502)(311,224,182,0,"c1",503,504,505)(222,315,224,0,"c1",506,507,508)(213,40,235,0,"c1",509,510,511)(198,184,31,0,"c0",512,475,513)(193,184,192,0,"c0",514,515,516)(186,185,302,0,"c0",517,518,519)(219,186,303,0,"c0",520,521,522)(269,189,188,0,"c1",523,524,525)(269,245,190,0,"c1",526,527,528)(246,245,269,0,"c1",529,526,530)(241,193,192,0,"c0",531,516,532)(219,303,304,0,"c0",522,533,534)(228,250,201,0,"c0",535,536,537)(247,195,249,0,"c0",538,539,540)(247,249,204,0,"c0",540,541,542)(215,197,196,0,"c1",543,544,545)(198,31,41,0,"c0",513,75,546)(41,199,198,0,"c0",547,548,546)(199,42,200,0,"c0",549,550,551)(43,201,200,0,"c0",552,553,554)(45,202,201,0,"c0",555,556,557)(45,203,202,0,"c0",558,559,555)(46,248,203,0,"c0",560,561,562)(48,55,205,0,"c1",94,563,564)(244,32,158,0,"c0",565,96,566)(244,123,32,0,"c1",567,99,565)(260,277,207,0,"c1",568,569,570)(36,35,254,0,"c1",105,571,572)(209,310,221,0,"c1",573,574,575)(239,240,38,0,"c1",576,577,578)(313,314,294,0,"c1",579,580,581)(185,186,273,0,"c1",517,582,497)(216,264,219,0,"c1",583,584,585)(187,223,216,0,"c1",586,502,587)(189,225,188,0,"c1",588,589,524)(189,190,224,0,"c1",590,591,592)(192,184,198,0,"c0",515,512,593)(185,177,302,0,"c0",499,495,518)(236,238,305,0,"c0",594,595,596)(210,188,194,0,"c1",597,598,599)(211,210,195,0,"c1",600,601,602)(195,196,211,0,"c1",603,604,602)(198,199,192,0,"c0",548,605,593)(199,200,192,0,"c0",551,606,605)(241,192,200,0,"c0",532,606,607)(203,228,202,0,"c0",608,609,559)(204,249,203,0,"c0",541,610,611)(205,206,196,0,"c1",612,613,614)(41,42,199,0,"c0",149,549,547)(43,200,42,0,"c0",554,550,150)(43,44,201,0,"c0",152,615,552)(44,45,201,0,"c0",153,557,615)(45,46,203,0,"c0",154,562,558)(47,248,46,0,"c0",616,560,156)(205,248,48,0,"c0",617,618,564)(260,34,33,0,"c1",619,159,620)(255,208,287,0,"c1",621,622,623)(274,126,265,0,"c1",624,625,626)(182,209,221,0,"c1",627,575,628)(182,183,209,0,"c1",629,630,627)(179,256,268,0,"c1",631,632,633)(38,52,231,0,"c1",176,634,635)(195,210,237,0,"c1",601,636,637)(246,269,210,0,"c1",530,638,639)(54,40,213,0,"c1",182,509,640)(55,206,205,0,"c1",641,612,563)(197,214,211,0,"c1",642,643,644)(197,211,196,0,"c1",644,604,544)(115,114,220,0,"c1",189,645,646)(214,212,245,0,"c1",647,648,649)(230,235,222,0,"c1",650,651,652)(191,213,212,0,"c1",653,654,655)(54,213,191,0,"c1",640,653,656)(214,191,212,0,"c1",657,655,647)(197,191,214,0,"c1",658,657,642)(206,215,196,0,"c1",659,545,613)(179,268,217,0,"c1",633,660,661)(216,299,233,0,"c1",662,663,664)(201,202,228,0,"c0",556,609,537)(248,205,247,0,"c0",617,665,666)(247,204,248,0,"c0",542,667,666)(219,304,238,0,"c0",534,668,669)(219,187,216,0,"c1",670,587,585)(222,220,295,0,"c1",671,672,673)(220,57,294,0,"c1",674,675,676)(181,221,293,0,"c1",677,484,485)(182,221,181,0,"c1",628,677,678)(212,213,230,0,"c1",654,679,680)(295,294,314,0,"c1",681,580,682)(245,212,230,0,"c1",648,680,683)(216,298,299,0,"c1",684,685,662)(225,312,223,0,"c1",686,500,687)(312,225,311,0,"c1",686,688,689)(223,188,225,0,"c1",690,589,687)(188,223,187,0,"c1",690,586,691)(224,311,225,0,"c1",503,688,692)(189,224,225,0,"c1",592,692,588)(190,222,224,0,"c1",693,508,591)(224,183,182,0,"c1",694,629,504)(226,234,261,0,"c1",695,696,697)(216,233,264,0,"c1",664,698,583)(252,306,218,0,"c0",699,700,701)(228,227,250,0,"c0",702,703,535)(218,307,237,0,"c0",704,705,706)(306,307,218,0,"c0",707,704,700)(280,135,76,0,"c2",708,252,709)(259,229,289,0,"c1",710,711,712)(229,75,37,0,"c2",491,258,488)(230,222,190,0,"c1",652,693,713)(213,235,230,0,"c1",511,650,679)(292,291,290,0,"c2",714,715,716)(239,292,183,0,"c1",717,718,719)(242,232,229,0,"c2",720,492,721)(178,261,243,0,"c0",722,723,724)(179,217,276,0,"c1",661,725,726)(226,261,263,0,"c1",697,727,728)(235,220,222,0,"c1",729,671,651)(115,220,235,0,"c1",646,729,730)(194,188,236,0,"c1",598,731,732)(188,187,236,0,"c1",691,733,731)(238,187,219,0,"c1",734,670,669)(203,249,228,0,"c0",610,735,608)(236,187,238,0,"c1",733,734,594)(305,252,270,0,"c0",736,737,738)(127,309,35,0,"c1",739,740,287)(239,38

,231,0,"c1",578,635,741)(183,292,209,0,"c1",718,742,630)(183,240,239,0,"c1",743,576,719)(39,38,240,0,"c1",291,577,744)(227,193,241,0,"c0",745,531,746)(250,227,241,0,"c0",703,746,747)(259,242,229,0,"c2",748,721,710)(244,273,272,0,"c1",749,750,751)(177,273,244,0,"c1",498,749,752)(244,158,300,0,"c0",566,753,754)(260,261,234,0,"c1",755,696,756)(245,230,190,0,"c1",683,713,527)(214,245,246,0,"c1",649,529,757)(211,214,246,0,"c1",643,757,758)(211,246,210,0,"c1",758,639,600)(247,196,195,0,"c0",759,603,538)(205,196,247,0,"c0",614,759,665)(203,248,204,0,"c0",561,667,611)(48,248,47,0,"c0",618,616,308)(307,249,237,0,"c0",760,761,705)(195,237,249,0,"c0",637,761,539)(201,250,200,0,"c0",536,762,553)(250,241,200,0,"c0",747,607,762)(297,271,256,0,"c1",763,764,765)(236,305,194,0,"c0",596,766,732)(270,316,194,0,"c1",767,768,769)(283,282,253,0,"c1",770,771,772)(207,296,284,0,"c1",773,774,775)(208,265,266,0,"c1",776,777,778)(208,266,287,0,"c1",778,779,622)(265,36,254,0,"c1",780,572,781)(207,256,296,0,"c1",782,783,773)(256,207,268,0,"c1",782,784,632)(179,257,256,0,"c1",785,786,631)(257,297,256,0,"c1",787,765,786)(179,298,257,0,"c1",788,789,785)(181,257,180,0,"c1",487,790,791)(218,237,258,0,"c1",706,792,793)(237,210,258,0,"c1",636,794,792)(209,242,259,0,"c2",795,748,796)(259,289,209,0,"c1",712,797,796)(260,262,261,0,"c0",798,799,755)(207,281,260,0,"c1",800,801,570)(243,261,262,0,"c0",723,799,479)(33,262,260,0,"c1",802,798,620)(233,275,263,0,"c1",803,804,805)(178,263,261,0,"c0",806,727,722)(272,123,244,0,"c1",807,567,751)(178,264,263,0,"c0",808,809,806)(264,233,263,0,"c1",698,805,809)(219,264,186,0,"c1",584,810,520)(264,178,186,0,"c1",808,811,810)(114,57,220,0,"c1",361,674,645)(40,115,235,0,"c1",362,730,510)(265,254,267,0,"c1",781,812,813)(208,274,265,0,"c1",814,626,776)(286,267,285,0,"c1",815,816,817)(266,267,286,0,"c1",818,815,819)(265,267,266,0,"c1",813,818,777)(217,268,251,0,"c1",660,820,821)(251,268,278,0,"c1",820,822,823)(188,210,269,0,"c1",597,638,525)(269,190,189,0,"c1",528,590,523)(270,252,316,0,"c1",737,824,767)(194,305,270,0,"c0",766,738,769)(271,253,256,0,"c1",825,826,764)(287,266,286,0,"c1",779,819,827)(272,33,123,0,"c1",828,379,807)(262,33,272,0,"c1",802,828,480)(273,243,272,0,"c1",829,481,750)(273,178,243,0,"c1",830,724,829)(186,178,273,0,"c1",811,830,582)(265,126,36,0,"c1",625,383,780)(87,309,127,0,"c1",831,739,385)(229,145,288,0,"c1",489,832,833)(289,229,288,0,"c1",711,833,834)(275,226,263,0,"c1",835,728,804)(299,179,276,0,"c1",836,726,837)(278,207,277,0,"c4",838,569,839)(260,234,277,0,"c1",756,840,568)(268,207,278,0,"c1",784,838,822)(280,76,279,0,"c2",709,841,842)(52,135,280,0,"c2",397,708,843)(280,231,52,0,"c2",844,634,843)(209,292,290,0,"c1",742,716,845)(281,34,260,0,"c1",846,619,801)(284,281,207,0,"c1",847,800,775)(281,284,34,0,"c1",847,848,846)(284,282,140,0,"c1",849,850,851)(283,141,282,0,"c1",852,853,770)(267,254,285,0,"c1",812,854,816)(140,34,284,0,"c1",410,848,851)(296,282,284,0,"c1",855,849,774)(282,141,140,0,"c1",853,412,850)(283,308,141,0,"c1",856,857,852)(308,283,285,0,"c1",856,858,859)(253,286,285,0,"c1",860,817,861)(285,283,253,0,"c1",858,772,861)(253,271,286,0,"c1",825,862,860)(271,287,286,0,"c1",863,827,862)(297,287,271,0,"c1",864,863,763)(254,309,285,0,"c1",865,866,854)(288,145,126,0,"c1",832,424,867)(126,274,288,0,"c1",624,868,867)(208,288,274,0,"c1",869,868,814)(289,288,208,0,"c1",834,869,870)(209,289,310,0,"c1",797,871,573)(291,279,290,0,"c2",872,873,715)(280,279,291,0,"c2",842,872,874)(291,231,280,0,"c2",875,844,874)(292,231,291,0,"c2",876,875,714)(239,231,292,0,"c1",741,876,717)(289,208,310,0,"c1",870,877,871)(255,297,293,0,"c1",878,879,483)(294,57,313,0,"c1",675,493,581)(220,294,295,0,"c1",676,681,672)(253,282,296,0,"c1",771,855,880)(296,256,253,0,"c1",783,826,880)(255,287,297,0,"c1",623,864,878)(293,297,257,0,"c1",879,787,486)(298,216,312,0,"c1",684,501,881)(180,257,298,0,"c1",790,789,882)(299,276,233,0,"c1",837,883,663)(298,179,299,0,"c1",788,836,685)(177,244,300,0,"c0",752,754,477)(158,303,300,0,"c0",443,884,753)(30,184,300,0,"c0",474,885,884)(184,301,300,0,"c0",886,478,885)(193,301,184,0,"c0",887,886,514)(302,301,193,0,"c0",496,887,888)(303,186,302,0,"c0",521,519,889)(193,303,302,0,"c0",890,889,888)(304,303,193,0,"c0",533,890,891)(193,227,304,0,"c0",745,892,891)(238,304,305,0,"c0",668,893,595)(307,228,249,0,"c0",894,735,760)(227,228,307,0,"c0",702,894,895)(307,306,227,0,"c0",707,896,895)(252,305,306,0,"c0",736,897,699)(305,227,306,0,"c0",898,896,897)(304,227,305,0,"c0",892,898,893)(254,35,309,0,"c1",571,740,865)(141,308,87,0,"c1",857,899,459)(285,309,308,0,"c1",866,900,859)(309,87,308,0,"c1",831,899,900)(255,310,208,0,"c1",901,877,621)(255,221,310,0,"c1",482,574,901)(311,181,180,0,"c1",902,791,903)(182,181,311,0,"c1",678,902,505)(312,180,298,0,"c1",904,882,881)(312,311,180,0,"c1",689,903,904)(313,172,39,0,"c1",494,465,905)(39,240,313,0,"c1",744,906,905)(240,314,313,0,"c1",907,579,906)(183,314,240,0,"c1"

ANNEXE 2

Cette annexe a pour but de donner une idée de ce que peut représenter la description géométrique tridimensionnelle d'une scène . Elle présente le contenu d'un fichier ASCII produit par le logiciel ED3D réalisé pendant la période de stage au LIG - EPFL (Lausanne 1990). Il comprend la liste des noeuds qui composent la hiérarchie arborescente de la scène. On peut remarquer que les noeuds "typés" sont accompagnés d'informations relatives au type (par exemple, "Camera Info").

SCENE NAME : scene3.scn
NUMBER OF NODES : 14
STATUS CAMERAS : 1
STATUS LIGHTS : 1
STATUS REPAIRS : 0

***** NODE *****

NUM NODE : 1
NODE NAME : first
NODE TYPE : 0
MATRIX : 0.9856 0.1407 -0.0937 0.0000
-0.0922 0.9121 0.3994 0.0000
0.1417 -0.3850 0.9120 0.0000
42.2841 -1.2174 -41.0770 1.0000
STATUS REPAIRS : 1
STATUS LINKS : 1
NUMBER OF CHILDS : 5

***** NODE *****

NUM NODE : 2
NODE NAME : Scene
NODE TYPE : 3
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
** CAMERA INFO **
CAMERA TYPE : 7
ZOOM : 1.0000
FIELD OF VIEW : 900.0000
ASPECT RATIO : 1.0000
RIGHT CLIPPING : 150.0000
LEFT CLIPPING : -150.0000
BOTTOM CLIPPING : -150.0000
TOP CLIPPING : 150.0000
NEAR PLANE : 1.0000
FAR PLANE : 1000.0000
TWIST : 0.0000
MATRIX : 0.7071 -0.4082 0.5774 0.0000
0.0000 0.8165 0.5774 0.0000
-0.7071 -0.4082 0.5774 0.0000
0.0000 0.0000 -173.2051 1.0000
LOOKING MODE : 1
ANGLE AROUND X : 0.5000
ANGLE AROUND Y : 0.5000
CAMERA-NODE DISTANCE : 500.0000
NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 3
NODE NAME : Light
NODE TYPE : 4
MATRIX : 1.0000 0.0000 0.0000 0.0000

0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
-4.1129 -25.8363 131.4725 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** LIGHT INFO ****
LIGHT TYPE : 1
COLOR : 0.0000 1.0000 0.0000
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
POSITION : 0.0000 0.0000 0.0000 1.0000
DIRECTION : 1
STATUS LIGHT : 1
NUMBER OF CHILDS : 1

******* NODE *******

NUM NODE : 4
NODE NAME : Figure
NODE TYPE : 5
MATRIX : 0.4395 -0.5342 -0.7221 0.0000
0.0622 0.8200 -0.5690 0.0000
0.8961 0.2051 0.3936 0.0000
-81.3802 80.8736 -114.1028 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** FIGURE INFO ****
SM_FIGURE NAME : elvis_a.sm
STAT_VISUAL : 5
NUMBER OF FATHERS : 1
NAME OF FATHER : Light
NUMBER OF CHILDS : 0

******* NODE *******

NUM NODE : 5
NODE NAME : Light
NODE TYPE : 4
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
-60.1798 103.9126 60.0434 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** LIGHT INFO ****
LIGHT TYPE : 1
COLOR : 1.0000 0.0000 0.0000
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
POSITION : 0.0000 0.0000 0.0000 1.0000
DIRECTION : 2
STATUS LIGHT : 1
NUMBER OF CHILDS : 1

******* NODE *******

NUM NODE : 6
NODE NAME : Figure
NODE TYPE : 5
MATRIX : 0.7658 0.6040 0.2204 0.0000
0.2888 -0.6293 0.7216 0.0000
0.5746 -0.4890 -0.6563 0.0000
-48.7624 -104.2494 65.7910 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** FIGURE INFO ****
SM_FIGURE NAME : pingouin.sm
STAT_VISUAL : 5
NUMBER OF FATHERS : 1
NAME OF FATHER : Figure
NUMBER OF CHILDS : 0

******* NODE *******

NUM NODE : 7
NODE NAME : Figure
NODE TYPE : 5
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 50.0000 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** FIGURE INFO ****
SM_FIGURE NAME : sphere20.sm
STAT_VISUAL : 5
NUMBER OF FATHERS : 2
NAME OF FATHER : first
NAME OF FATHER : first
NUMBER OF CHILDS : 0

******* NODE *******

NUM NODE : 8
NODE NAME : Light
NODE TYPE : 4
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
19.1662 -44.3812 102.6064 1.0000
STATUS REPAIRS : 0
STATUS LINKS : 1
**** LIGHT INFO ****
LIGHT TYPE : 1
COLOR : 0.0000 0.0000 1.0000
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
POSITION : 0.0000 0.0000 0.0000 1.0000
DIRECTION : 2
STATUS LIGHT : 1

NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 9

NODE NAME : first

NODE TYPE : 0

MATRIX : 0.5522 -0.8328 -0.0403 0.0000

0.3122 0.2513 -0.9162 0.0000

0.7731 0.4933 0.3987 0.0000

14.0308 -1.1958 -12.8565 1.0000

STATUS REPAIRS : 1

STATUS LINKS : 1

NUMBER OF CHILDS : 5

***** NODE *****

NUM NODE : 10

NODE NAME : Scene

NODE TYPE : 3

MATRIX : 1.0000 0.0000 0.0000 0.0000

0.0000 1.0000 0.0000 0.0000

0.0000 0.0000 1.0000 0.0000

0.0000 0.0000 0.0000 1.0000

STATUS REPAIRS : 1

STATUS LINKS : 1

** CAMERA INFO **

CAMERA TYPE : 7

ZOOM : 1.0000

FIELD OF VIEW : 900.0000

ASPECT RATIO : 1.0000

RIGHT CLIPPING : 150.0000

LEFT CLIPPING : -150.0000

BOTTOM CLIPPING : -150.0000

TOP CLIPPING : 150.0000

NEAR PLANE : 1.0000

FAR PLANE : 1000.0000

TWIST : 0.0000

MATRIX : 0.7071 -0.4082 0.5774 0.0000

0.0000 0.8165 0.5774 0.0000

-0.7071 -0.4082 0.5774 0.0000

0.0000 0.0000 -173.2051 1.0000

LOOKING MODE : 1

ANGLE AROUND X : 0.5000

ANGLE AROUND Y : 0.5000

CAMERA-NODE DISTANCE : 500.0000

NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 11

NODE NAME : Figure

NODE TYPE : 5

MATRIX : 0.4070 0.8518 0.3297 0.0000

0.9093 -0.3431 -0.2359 0.0000

-0.0877 0.3959 -0.9141 0.0000

8.8414 -65.4508 -13.4217 1.0000

STATUS REPAIRS : 1

STATUS LINKS : 1
** FIGURE INFO **
SM_FIGURE NAME : pingouin.sm
STAT_VISUAL : 5
NUMBER OF FATHERS : 2
NAME OF FATHER :first
NAME OF FATHER :first
NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 12
NODE NAME : Light
NODE TYPE : 4
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
50.0000 0.0000 0.0000 1.0000
STATUS REPAIRS : 1
STATUS LINKS : 1
** LIGHT INFO **
LIGHT TYPE : 0
COLOR : 0.0000 1.0000 0.0000
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
POSITION : 0.0000 0.0000 0.0000 1.0000
DIRECTION : 1
STATUS LIGHT : 1
NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 13
NODE NAME : Light
NODE TYPE : 4
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
50.4254 91.7391 -86.4243 1.0000
STATUS REPAIRS : 1
STATUS LINKS : 1
** LIGHT INFO **
LIGHT TYPE : 1
COLOR : 1.0000 0.0000 0.0000
MATRIX : 1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000
0.0000 0.0000 0.0000 1.0000
POSITION : 0.0000 0.0000 0.0000 1.0000
DIRECTION : 2
STATUS LIGHT : 1
NUMBER OF CHILDS : 0

***** NODE *****

NUM NODE : 14

NODE NAME : Figure

NODE TYPE : 5

MATRIX : -0.5407 -0.7698 0.3393 0.0000

-0.8124 0.729 -0.4484 0.0000

0.2186 -0.5182 -0.8268 0.0000

9.2393 94.0300 17.8704 1.0000

STATUS REPAIRS : 1

STATUS LINKS : 1

** FIGURE INFO **

SM_FIGURE NAME : elvis_a.sm

STAT_VISUAL : 4

NUMBER OF FATHERS : 2

NAME OF FATHER :first

NAME OF FATHER :first

NUMBER OF CHILDS : 0

***** END OF SCENE *****

